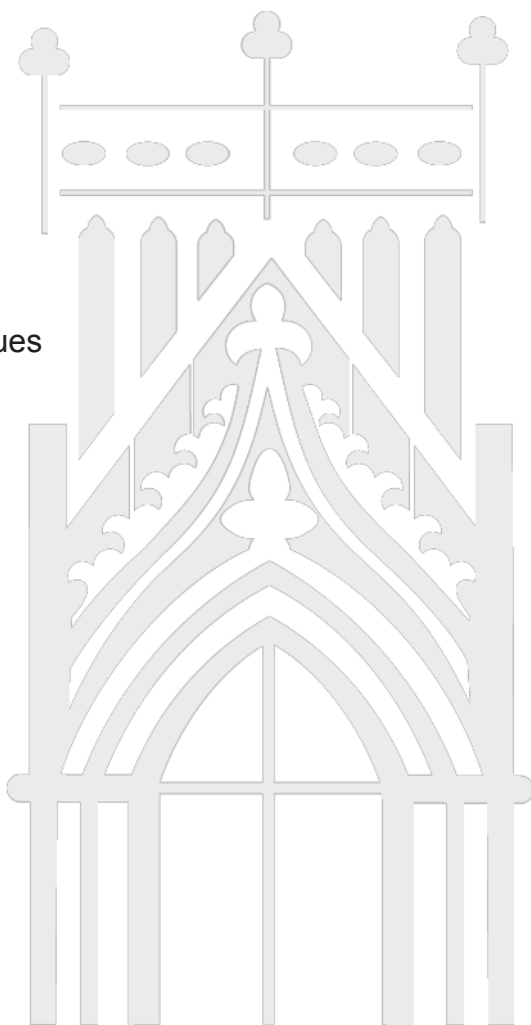


Mestrado em Computação Móvel

CPE Management e Backoffice

Paulo Jorge Domingues Rodrigues Marques

março | 2020



Escola Superior
de Tecnologia e Gestão



Escola Superior de Tecnologia e Gestão
Instituto Politécnico da Guarda

CPE Management e Backoffice

Relatório de Projeto Aplicado submetido como requisito parcial para
obtenção do grau de Mestre em Computação Móvel

Orientador: Professor José Fonseca

Paulo Jorge Domingues Rodrigues Marques

Março | 2020

[O mais difícil não é escrever muito: é dizer tudo, escrevendo pouco.]

[Júlio Dantas.]

[Aos meus pais.]

Agradecimentos

[Este projeto foi o culminar de um conjunto de fatores que permitiram terminar com sucesso, dos quais destaco a oportunidade que a administração da empresa Atena T me proporcionou ao entregar-me de forma incondicional a liderança e implementação, o apoio que a minha família, nomeadamente a Adelina, Daniela e Beatriz, e não esquecendo a disponibilidade dos professores Rui Pereira e José Fonseca.]

Resumo

A Atena T, desenvolve a sua atividade na área das telecomunicações, passou a pertencer a um grupo restrito de fornecedores de um dos mais importantes operadores de telecomunicações a operar em Portugal, a Vodafone. Passou a prestar serviços de instalações de ADSL e fibra, usualmente denominados de *Triple play* (Internet, TV e telefone).

A Vodafone apenas aceita novos fornecedores destes serviços na condição que estes apresentem meios de gestão de *stocks* do material entregue à consignação, que é feita com recurso a um *web service* que é consumido através de dois métodos, *Consign Stock* e *Check Consignment Status*. O *Consign Stock* tem como objetivo alocar os equipamentos ao cliente onde foi efetuada a instalação do serviço, que podem ser *router's*, *Tv Box*, telefones, comandos, cabos e tomadas, e o *Check Consignment Status* permite verificar o estado do *Consign Stock*. A troca de informação efetuada através do *web service* tem por base os documentos em linguagem UBL (*Universal Business Language*).

Para iniciar este fornecimento de serviço, a empresa Atena T procedeu à elaboração de uma aplicação para consumir o *web service* da Vodafone. Optou-se pelo desenvolvimento *in house* de um protótipo do tipo *desktop* em JAVA que cumpre o solicitado pela Vodafone. Isto permitiu ganhar tempo para desenvolver um portal *web* que, para além de consumir o *web service*, apresenta outras funcionalidades como disponibilizar as ordens de serviço aos técnicos, que também passaram a ter acesso ao serviço através de uma aplicação para *smartphone* desenvolvida no decorrer deste projeto.

As aplicações desenvolvidas foram alvo de testes por parte da Vodafone em dois ambientes, teste e produção, e em ambos os casos os testes foram realizados com sucesso. Durante cerca de um ano em que a prestação de serviço ocorreu, as aplicações foram utilizadas pelos funcionários da Atena T. Apesar da prestação de serviços ter terminado as aplicações poderão ser utilizadas futuramente em cenários semelhantes.

Palavras-chave: Operadores de telecomunicações, *Web service*, *Backoffice*, dispositivos móveis.

Abstract

Atena T, develops its activity in the area of telecommunications, start belonging to a restricted group of suppliers of one of the most important telecommunications operators operating in Portugal, Vodafone. Start providing services of ADSL and fiber facilities, usually called Triple play (Internet, TV and telephone).

Vodafone only accepts new suppliers of these services on the condition that they present means of managing the stocks of the material delivered to the consignment, which is made using a web service that is consumed through two methods, Consign Stock and Check Consignment Status. The Consign Stock aims to allocate the equipment to the customer where the service was installed, which can be the router's, Tv Box, telephones, commands, cables and sockets, and the Check Consignment Status allows checking the status of the Consign Stock. The exchange of information through the web service is based on UBL (Universal Business Language) language documents.

To start this service provision, the company Atena T has developed an application to consume the web service. The in-house development of a prototype of the desktop type in JAVA that complies with what Vodafone has requested has allowed us to gain time to develop a web portal that, in addition to consuming the web service, presents other functionalities such as making service orders available to technicians, who also had access to the service, through a smartphone application developed during this project.

The applications were tested by Vodafone in two environments, test and production, and in both cases were successfully performed.

For about 1 year the applications were used by the employees of Atena T. Although the service rendering has finished the applications may be used in the future.

Key Words: Aplication, Web service, Backoffice, Plataform, mobile devices.

Índice

Índice de Figuras	vi
Siglas	ix
1. Introdução	1
1.1. Contexto e motivação	1
1.2. Identificação do problema.....	2
1.3. Objetivos.....	3
1.4. Implementação do projeto	5
1.5. Estrutura do relatório	5
2. Estado da Arte	7
2.1. Aplicações prontas a usar.....	7
2.2. Consultoras de <i>outsourcing</i>	8
2.3. Desenvolvimento interno	9
3. Especificação do <i>Web service</i> da Vodafone	13
3.1. Objetivo	13
3.2. Descrição	13
3.3. Modelo de informação	14
3.4. UBL 2.0 <i>Receipt Advice</i>	14
3.5. UBL 2.0 <i>Application Response</i>	16
3.6. Componente de ligação.....	16
3.7. <i>Consign Stock</i>	17
3.8. <i>Check Consignment Status</i>	18
3.9. Tipo de dados.....	19
4. <i>Universal Business Language</i>	21
4.1. Enquadramento da UBL no projeto	21
4.2. O que é a UBL?	21
4.3. Implementação da UBL no projeto.....	23
5. Desenvolvimento da <i>deskCPE</i> em JAVA	33
5.1. Segurança.....	38
5.2. Testes	39
5.3. Distribuição.....	43
6. Parametrização do ERP PHC	47
7. Desenvolvimento do <i>WEBCPE</i>	53
7.1. Ferramentas de desenvolvimento.....	53
7.2. Estrutura do <i>webCPE</i>	53
7.2.1. A área pública.....	54
7.2.2. Zona reservada.....	56
7.3. Aplicação <i>Web</i>	61
7.4. Integração do <i>web service</i> no projeto	62
7.4.1. Método 1 - <i>Add-reference</i>	62
7.4.2. Método 2 - <i>Add-Service reference (Advanced)</i>	63
7.4.3. Método 3 - <i>Add-Service reference (Advanced) 2</i>	64
7.4.4. Método 4 - <i>WDSL.exe</i>	64

7.5.	Sistemas de base de dados e <i>profiling</i>	65
7.6.	Segurança do sistema.....	72
8.	Aplicação para dispositivos móveis (<i>Smartphones</i>).....	77
8.1.	<i>Titanium Studio</i>	80
8.2.	<i>Multi-device Hybrid Apps for Visual Studio</i>	81
8.3.	PhoneGap.....	81
8.4.	Intel XDK	82
8.5.	<i>Responsive Website</i> com bootstrap	84
9.	Conclusão	87
	Referências Bibliográficas.....	89
	Anexo A – Classes Desenvolvidas	91
	Anexo B – Testes da Aplicação.....	129

Índice de Figuras

Figura 1. <i>Receipt Advice</i>	15
Figura 2 - <i>Application Response</i>	16
Figura 3. <i>Consign Stock</i>	17
Figura 4. <i>Check Consignment Status</i>	18
Figura 5 – Estrutura do <i>ReceiptAdvice</i>	23
Figura 6 - Código XML <i>Receipt Advice</i>	25
Figura 7 - Código XML <i>Response</i>	26
Figura 8 - Código XML <i>DocumentReference</i>	27
Figura 9 - Código XML da <i>Application Response</i>	27
Figura 10 - Fluxograma <i>ConsignStock</i>	29
Figura 11 - Fluxograma <i>CheckConsignmentStatus</i>	30
Figura 12 – Classes Geradas.....	34
Figura 13 - Classes desenvolvidas.....	34
Figura 14 - Integração métodos	36
Figura 15 – Painel <i>Consign</i>	37
Figura 16 - Painel <i>check status</i>	38
Figura 17 - Software Launch4J	45
Figura 18 - PHC Dossier Interno.....	48
Figura 19 - PHC Acesso parametrização dossier	48
Figura 20 - PHC Menu de parametrização dossier.....	49
Figura 21 - PHC Acesso Campos de utilizador.....	49
Figura 22 – PHC Painel Campos de utilizador.....	50
Figura 23 - PHC Configuração nº série no Dossier.....	51
Figura 24 - PHC Ativar nº Série em Stock e serviços	51
Figura 25 - PHC Janela introdução nº série.....	52
Figura 26 - Página <i>Home</i>	54
Figura 27 - Página de contactos.....	55
Figura 28 - Página de informação sobre a empresa.....	55
Figura 29 - Acesso área reservada.....	56
Figura 30 - Página "área".....	57
Figura 31 - Página "Técnicos"	58
Figura 32 - Página "Armazém"	58
Figura 33 – Página “Envio WO”	59
Figura 34 - Página " <i>Checkstatus</i> "	60
Figura 35 - Página " <i>ReceiptLine</i> "	61
Figura 36 - Add service reference Https.....	63
Figura 37 – Add service reference File WSDL	64
Figura 38 - Tabelas base dados CPE	66
Figura 39 - Tabelas Base dados <i>Webservice</i>	66
Figura 40 - <i>Express Profiler Login</i>	68
Figura 41 – <i>Express profiler output</i>	69
Figura 42 - <i>Express profiler query</i>	70

Figura 43 - Filtros <i>Express Profiler</i>	71
Figura 44 - <i>Asp.Net Configuration</i>	72
Figura 45 - Gestão de <i>roles</i> e <i>users</i>	73
Figura 46 - Encriptação	75
Figura 47 - Interface XDK.....	83
Figura 48 - XDK Link to email	84

Siglas

CPE – Customer Premises Equipment

ERP – Enterprise Resource Planning

FAQ - frequently asked questions

FSC – Field Service Company

FSS – Field Service Supplier

IDE - Integrated Development Environment

JAR - Java ARchive

OASIS – Advancing Open Standards for the Information Society

SAP – Systems, Applications and Products in Data Processing

SQL - Structured Query Language

UBL – Universal Business Language

WSDL-Web Service Definition Language

XML - eXtensible Markup Language

1. Introdução

Este primeiro capítulo tem como objetivo dar a conhecer o contexto e as motivações deste projeto desenvolvido no seio da empresa Atena T, identificar o problema a apresentar soluções a serem desenvolvidas. Este projeto reveste-se de um nível de exigência elevado porque estamos perante a Vodafone, um dos maiores operadores de telecomunicações a operar a nível mundial (Investopedia, 2019), o que implica que seja necessário elaborar as aplicações com muito cuidado, profissionalismo e com objetividade. Será necessário criar o *backoffice* que irá produzir a documentação em formato UBL-*Universal Business Language*¹ (UBL, 2016) e consequente interface com o sistema de ERP (*Enterprise Resource Planning*) da Atena T.

Este projeto apresenta-se como uma oportunidade de desenvolver competências na área de programação em âmbito profissional. A proposta de desenvolver uma aplicação com os recursos da Atena T é de longe a mais vantajosa, a administração deu luz verde.

1.1. Contexto e motivação

Os operadores de telecomunicações utilizam a designação técnica CPE (*Customer Premises Equipment*) para se referirem ao equipamento que se encontra entregue ao cliente. A gestão destes ativos do operador é designada por CPE *Management*. Os processos envolvidos nesta gestão podem ser novas instalações de serviços e os respetivos equipamentos, trocas de equipamentos nas instalações do cliente por avaria ou *upgrade* de equipamento, aquisição por parte do cliente de serviços adicionais e desinstalação de serviços.

Nestes processos é muito importante que a informação seja a mais fiável possível, e os equipamentos devem ser movimentados através do seu ID (número de série ou *MAC address*). Os equipamentos envolvidos na operação devem estar sempre

¹ UBL é uma linguagem baseada em XML que é utilizada nas trocas de documentos entre empresas. Estes são, a título de exemplo, as faturas e guias de remessa.

localizáveis, ou seja, o *Backoffice* que controla as instalações de serviços deve saber que equipamentos possui cada técnico para poder alocar as ordens de serviço.

Devido ao elevado custo dos equipamentos, o *stock* disponibilizado pelo operador é calculado de acordo com o serviço previsto semanalmente. Desta forma, o departamento de logística das equipas técnicas que fazem as instalações, como é o caso da Atena T, tem de estar em contacto com o *backoffice* para que apenas seja entregue o indispensável aos técnicos, com uma pequena margem de segurança.

1.2. Identificação do problema

Numa reunião entre a Atena T e a Vodafone é dado a conhecer o projeto global, no qual a Vodafone pretende incluir no seu grupo de fornecedores a Atena T como instalador de serviços *Triple Play*. A seleção de fornecedores nesta área é feita tendo em atenção o nível de qualidade de serviço, bem como o prestígio reconhecido no mercado.

A Atena T tem de dotar a sua estrutura com meios suficientes, que são de carácter técnico, humano e financeiro, para responder aos pedidos do seu cliente Vodafone. Será necessário criar, organizar e coordenar equipas técnicas para executar as instalações de TV/Net/Voz, constituir um grupo para efetuar as tarefas de *backoffice* e logística. No entanto, existe um requisito sem o qual o projeto não pode passar à prática, que é o desenvolvimento de uma ferramenta aplicacional que interaja com um *web service* da Vodafone, denominado *ConsignStock*. Este *web service* disponibiliza unicamente duas funções:

1. Submeter as ordens de serviço efetuadas pelos técnicos. Para tal, tem de ser enviado um documento XML (*eXtensible Markup Language*) em linguagem UBL do tipo *receipt advice*. Este descreve detalhadamente que equipamentos foram entregues ao cliente no momento da instalação do serviço TV/Net/voz. Esta função faz uma verificação imediata de alguns elementos que constam no *receipt advice* e retorna uma mensagem de sucesso ou de erro detalhado.
2. Verificar qual o estado das submissões de ordens de serviço. O objetivo é enviar um documento UBL do tipo *application response*. Neste documento consta o número da ordem de serviço que se pretende obter e a confirmação se os equipamentos já se encontram

alocados ao cliente, ou seja, se já deixaram de estar sob a responsabilidade do instalador (neste caso, da Atena T). Também se obtem uma resposta do sistema que indica qual o estado e, se existirem, indicação dos erros encontrados.

A Vodafone disponibilizou um *dossier* denominado “CPE Management – Especificação de *Web Services*” com a especificação do *web service*. Este contém informação bastante resumida, mas é suficientemente objetiva e clara, e é composto pelos seguintes tópicos:

- Introdução;
- Objetivo;
- Descrição do funcionamento do *web service*;
- Modelo de informação;
- UBL – documentos utilizados e diagramas estruturais;
- Componente de ligação;
- Funções disponibilizadas;
- Tipo de dados;

Que é descrito detalhadamente no 3º capítulo deste relatório.

Não estando disponível para a Atena T uma ferramenta de acesso ao *web service* da Vodafone, ela terá de ser desenvolvida, para boa execução do contrato. A responsabilidade deste desenvolvimento aplicacional passou para o colaborador da Atena T que ficou responsável por implementar uma solução num curto espaço temporal, que não poderá exceder os seis meses por imposição da Atena T e Vodafone.

1.3. Objetivos

O serviço de TV/NET/Voz disponibilizado pela Vodafone após ser contratado pelo cliente é então iniciado o processo de instalação, a Vodafone processa, no seu sistema o contrato do cliente. Posteriormente passa a informação ao instalador, neste caso a Atena T para proceder à instalação do serviço nas instalações do cliente.

Os operadores de *Backoffice* da Atena T recebem a informação das instalações de serviços que os seus técnicos tem de efectuar, esta é introduzida no sistema da Atena T. Quando os técnicos finalizam as instalações é necessário informar a

Vodafone do resultado das instalações, com indicação dos equipamentos aplicados na casa do cliente.

Após a entrega por parte da Vodafone da especificação do *web service* e respetivo ficheiro WSDL (*Web Services Description Language*), foi feita uma primeira avaliação por parte da Atena T. Concluiu-se que uma solução integrada, que implicaria desenvolver um sistema que efetuasse integração direta no ERP existente na Atena T, o PHC, e que teria de cumprir os requisitos internos dos diversos departamentos ligados ao projeto, iria demorar demasiado tempo e exceder o prazo de seis meses. Para agilizar a resolução desta necessidade de ter uma ferramenta que permitisse interagir com o *web service* da Vodafone em tempo útil, optou-se por promover o desenvolvimento rápido de uma aplicação mais pequena que não tivesse as condicionantes de uma aplicação completamente integrada.

A aplicação será disponibilizada numa fase inicial localmente nos computadores através de um executável. Após a identificação do tipo de aplicação escolheu-se o JAVA como linguagem de desenvolvimento, usando o *Netbeans* como IDE, de acordo com sugestões do cliente Vodafone.

De uma forma resumida, a aplicação a desenvolver deverá enviar um conjunto de informações através da invocação do *web service* da Vodafone. Este irá espoletar um conjunto de funções que efetuam as movimentações de *stocks* de equipamentos que se encontram consignados à Atena T no sistema de ERP SAP (*Systeme, Anwendungen und Produkte in der Datenverarbeitung*) da Vodafone.

A gestão dos equipamentos que são disponibilizados à Atena T pela Vodafone é efetuada através de uma ferramenta ERP que a Atena T já possui, o PHC. O que se pretende é interligar este sistema com um portal *web* que vai ser desenvolvido. Este portal irá possuir uma base de dados própria que servirá de ferramenta de trabalho para as equipas técnicas e *backoffice*. O portal *web* será o elemento mais importante deste projeto, pois irá compilar, produzir e enviar a documentação necessária para alimentar o *web service* da Vodafone para que a integração da informação seja perfeita.

Como será de esperar num projeto desta dimensão, não se deve esquecer que obrigatoriamente será necessário que o portal *web* a desenvolver permita que seja utilizada por dispositivos móveis. Esta questão é fundamental quando nos referimos às

equipas técnicas que terão necessidade de consultar as ordens de serviço através de um *smartphone* ou *tablet*.

1.4. Implementação do projeto

O projeto será desenvolvido tendo em atenção as seguintes fases:

1. Desenvolvimento de uma aplicação em JAVA denominada *deskCPE*, que será fundamental para que a Atena T possa desenvolver a sua atividade perante o operador Vodafone, sem estar dependente do *software* de um concorrente direto.
2. Parametrização do ERP PHC. Este sistema, na sua configuração base, não possui alguns elementos que são necessários para este projeto, como por exemplo um campo com o número da WO. Torna-se então fundamental proceder a alterações do PHC.
3. Desenvolvimento de portal *WEB* denominado *webCPE*. Esta fase trata-se de uma evolução do *deskCPE*.
4. Desenvolvimento de plataforma para *smartphone* denominada *smartCPE* que será utilizada pelas equipas técnicas.

Estes pontos são detalhados nos capítulos 5, 6, 7 e 8.

1.5. Estrutura do relatório

Este relatório está dividido em 9 capítulos. No capítulo 1 é feita a introdução, a contextualização, indicadas as motivações e descrita a solução apresentada. O capítulo 2 foca-se no estado da arte. No capítulo 3 é feita a descrição pormenorizada do *web service*. O capítulo 4 foca-se na linguagem UBL. O capítulo 5 descreve o desenvolvimento de aplicação *desktop* em *Java*. No capítulo 6 é feita descrição de parametrização do PHC. O capítulo 7 é dedicado ao desenvolvimento de portal *web*. O capítulo 8 é focado nas soluções no desenvolvimento de aplicações móveis consideradas neste projeto e no capítulo 9 é feita a conclusão.

O próximo capítulo é dedicado ao estudo do estado da arte

2. Estado da Arte

Neste capítulo é feita uma análise das soluções existentes no mercado com vista à resolução do problema, que dará resposta à exigência da Vodafone. O desafio proposto por este projeto tem três caminhos possíveis:

1. Aplicações prontas a usar. Procurar um produto já desenvolvido que se adapte a esta realidade.
2. Consultoras de *outsourcing*. Estas empresas desenvolvem *software* feito à medida.
3. Equipa interna de desenvolvimento de *software*. De seguida irão ser analisadas estas três possibilidades, as suas vantagens e inconvenientes.

2.1. Aplicações prontas a usar

A administração da Atena T, perante a informação disponibilizada pela Vodafone e tendo em atenção que não existe qualquer apoio técnico no desenvolvimento por parte deles, chegou à conclusão que devia fazer uma consulta ao mercado de *software* pronto a utilizar. No entanto, como era espectável, este tipo de aplicações não se encontra disponível pois as especificações contêm um conjunto de requisitos muito específicos que não são comuns com os solicitados habitualmente pelas PMEs. Como já se suspeitava, os programas que estão disponíveis no mercado são vocacionados para gestão financeira e comercial que se focam em necessidades habituais, tendo normalmente apenas em atenção o que é exigido por lei.

Por outro lado, as *software houses* desenvolvem maioritariamente aplicações de gestão, contabilidade e, em alguns casos, destinadas a mercados específicos como clínicas, agências imobiliárias entre outras. No entanto, para este mercado das telecomunicações com integração de *web service* externos não existem soluções prontas.

A solução aplicacional existente que poderia ser utilizada é propriedade exclusiva de um concorrente da Atena T, tendo em atenção que foi desenvolvida exclusivamente para ele. Foram realizadas várias tentativas de diálogo no sentido de

adquirir licenciamento do *software*, mas não se chegou a nenhum acordo viável entre as duas empresas.

2.2. Consultoras de *outsourcing*

Existem em Portugal inúmeras empresas de consultadoria na área do desenvolvimento de *software* (inforempresas, 2016). Na última década assistimos a um crescimento exponencial deste tipo de empresas, pois o mercado do desenvolvimento de aplicações à medida é sinónimo de faturação acima da média (hatchapps.com, 2019), tendo em atenção que são soluções que não estão disponíveis no mercado e as empresas que as solicitam acabam por ter de aceitar o custo que lhes é proposto visto a ausência de outras soluções mais em conta. Estas consultoras normalmente trabalham para grandes empresas nacionais e internacionais. Para apresentar à Atena T a melhor solução na implementação deste projeto foram efetuadas algumas consultas ao mercado através de um conjunto de contactos que a Antena T possuía. No entanto, estas consultas não produziram o resultado desejado.

A Vodafone permitiu que a Atena T iniciasse a prestação de serviços de instalação de ADSL com o recurso ao K1, um *software* de uma empresa concorrente que já trabalha com a Vodafone. O K1 é completamente autónomo de qualquer outro *software* e pode ser interligado com qualquer tipo de *software* de gestão ou ERP, de acordo com informação prestada pela empresa que o desenvolveu. Ele foi desenvolvido em várias fases ao longo de cinco anos, demonstrando assim tratar-se de um produto robusto com provas dadas no terreno. Esta apresenta-se como uma solução bastante válida, pois o pretendido é uma aplicação que permita comunicar com o *web service* da Vodafone e ligar com o ERP PHC da Atena T e é precisamente isso que o K1 faz.

A consultora *Knowledge Works* que desenvolveu o K1 demonstrou interesse em comercializar o K1, no entanto como este foi desenvolvido exclusivamente para um único cliente, seria necessário obter autorização junto dele para avançar com a compra. Neste caso, o valor base estimado seria de 60.000 € acrescido de 15.000 € para que fosse adaptado para a realidade da Atena T. Foi equacionada a aquisição do K1, pois o produto já se encontra a funcionar na realidade pretendida, no entanto tal não foi possível. A empresa que desenvolveu o *software* não apresentou proposta final porque provavelmente não conseguiu obter autorização junto da empresa proprietária dos direitos do K1.

Adicionalmente a consultora propôs, como alternativa, desenvolver uma solução de raiz que teria um custo projetado entre os 70.000 € e 100.000 €, a acrescer um custo anual de manutenção mínima de 14.000 €. Esta solução foi colocada de parte pois não existe da parte da Atena T orçamento disponível para o valor pretendido.

Outra solução usando *outsourcing* seria desenvolver um módulo dentro do ERP PHC que permita a comunicação direta com o *web service* da Vodafone. Assim haverá um sistema único e integrado a gerir todo o processo. O PHC é composto por módulos: gestão, contabilidade, pessoal, imobilizado, recursos humanos, web, documentos e frota. Sendo assim, o desejável seria desenvolver um módulo para o PHC que teria a grande vantagem de ser colocado num sistema de fácil aceitação para os utilizadores finais, pois já se encontram familiarizados com este *software*. O PHC é um sistema fechado que se encontra apenas acessível aos denominados “parceiros PHC”. Estes parceiros prestam serviços de assistência e parametrização do sistema consoante as necessidades das empresas.

Foi solicitada uma proposta de desenvolvimento à medida ao parceiro PHC que dá suporte à Atena T. O processo iniciou-se por um levantamento de requisitos, tendo posteriormente passado por vários pedidos de esclarecimentos e após três meses sem que houvesse uma proposta abandonou-se esta solução, o que obrigou a prolongar o prazo de implementação. De acordo com informação obtida mais tarde, esta empresa não apresentou uma proposta visto que não possui recursos suficientes para dar resposta a este tipo de projeto.

Foram ainda consultadas mais duas empresas que efetuam parametrizações em PHC, mas o resultado foi idêntico.

2.3. Desenvolvimento interno

Esta solução poderá ser mais versátil, pois permite alguma flexibilidade tendo em atenção que o contacto com o cliente Vodafone é muito próximo. Como vantagem adicional, para a Atena T, é o acesso ao código fonte e ferramentas de desenvolvimento, o que permite efetuar *upgrades* de desenvolvimento ou correções à medida.

Quando as empresas possuem recursos humanos com competências no desenvolvimento de *software* e nas suas capacidades de adaptação a novas realidades que podem advir deste projeto, como utilização de novas ferramentas ou tecnologias, faz todo

o sentido recorrer a estes. De facto, é possível reduzir o investimento da Atena T significativamente, já que não terá custos adicionais porque o responsável pela administração de sistemas, infraestrutura informática e *helpdesk* demonstrou a sua disponibilidade para abraçar este projeto e conciliá-lo com as restantes tarefas que desempenha.

Como ferramenta de desenvolvimento, foi utilizado inicialmente o *Netbeans* que é gratuito, não havendo, portanto, investimento por parte da Atena T. Com esta ferramenta foram dados os primeiros passos no desenvolvimento do portal *web*. Nas fases seguintes foi utilizado o *Visual Studio 2013* e o *SQL Server 2012* porque estas ferramentas apresentam-se como mais versáteis para o desenvolvimento pois possuem componentes que permitem que o desenvolvimento seja mais ágil, como por exemplo a utilização combinada de uma *GridView* com um *SQLDataSource*, o *SQLDataSource* é uma componente que permite efetuar uma query à base dados e quando ligada a uma *GridView*, esta componente mostra o resultado da query (ForumsASPNet, 2020). Estas ferramentas são da *Microsoft* e na sua versão *express* podem ser utilizadas de forma gratuita. No entanto optou-se por adquirir as versões profissionais, visto que têm mais funcionalidades e a opção de desenvolver o *software* com recursos internos permitiu esse investimento.

Para desenvolvimento da aplicação para *smartphone* foi feita uma pesquisa intensa para encontrar a melhor ferramenta para desenvolvimento tendo em conta que se pretende que a aplicação seja multiplataforma para ser utilizada nos diversos ambientes moveis mais expressivos, *Android*, *iPhone* e *Windows Phone*. Entre as soluções encontradas, *XDK*, *titanium* e *xamarin*, destacou-se uma ferramenta da *Intel* denominada *XDK*. A decisão pelo *XDK* revelou-se a melhor solução porque permite o desenvolvimento multiplataforma e dentro deste segmento apresenta-se como sendo o mais estável, após serem utilizadas as ferramentas referidas com as aplicações de *demo* e *sample*, disponibilizadas pelas entidades que disponibilizam as ferramentas. A *Intel* disponibiliza um conjunto de elementos de apoio como documentos, artigos, *FAQs*, *quick help*, tutoriais, *webinars* e outros recursos (Intel, 2020). Todos estes elementos de apoio foram um fator decisivo.

Do ponto de vista da administração e direção financeira da Atena T, o desenvolvimento por equipa interna é o mais interessante. No entanto, enquanto

responsável pelo projeto e desenvolvimento das aplicações, existem vários desafios que têm de ser superados:

1. Falta de experiência em projetos.
2. Falta de apoio técnico.
3. Impossibilidade de dedicação exclusiva ao projeto.
4. Existência de tecnologias novas que têm de ser desbravadas, tais como a integração de *web services* externos, desenvolvimento multiplataforma para dispositivos móveis e linguagem UBL, com a qual não havia conhecimento prévio.

A especificação do *web service* da Vodafone é o próximo passo a seguir.

3. Especificação do *Web service* da Vodafone

O levantamento de requisitos de qualquer projeto é fundamental para conseguir atingir o objetivo com sucesso. Conforme já foi referido o ponto de partida é um *web service* disponibilizado pela Vodafone. Neste capítulo é apresentado o seu funcionamento, visto ser crucial o seu conhecimento detalhado para este projeto.

3.1. Objetivo

O objetivo principal deste projeto é ligar o sistema de SAP da Vodafone com o *software* da Atena T. O sistema da Vodafone permite receber informação referente aos clientes e movimentar os *stocks*. A Atena T irá efetuar instalação de serviços da Vodafone em casa dos clientes e, quando terminado o trabalho, a alocação de equipamento deve ser enviada para a Vodafone. Este procedimento é feito ao final do dia. O *software* da Atena T irá solicitar à Vodafone a transferência dos equipamentos do seu armazém para o cliente.

3.2. Descrição

O funcionamento do *web service* tem como elemento principal o *ADSL Id* ou *Access Id*, abreviado como ID, que indica qual o cliente e local de instalação do serviço. Sempre que o *web service* é invocado, o *software* da Atena T deve, sem exceção, enviar o ID pois, se este não constar, o *web service* devolve imediatamente uma mensagem de erro. Podem existir vários IDs para o mesmo cliente. Esta situação acontece quando um cliente subscreve vários serviços, e o caso típico será uma empresa de retalho que possua várias lojas. Toda a troca de informação com o *web service* é efetuada por elementos construídos em linguagem XML.

A Atena T faz a gestão de *stock* no seu ERP PHC. O técnico, após conclusão da intervenção no cliente, inicia um processo denominado *consign stock*, na aplicação da Atena T, que será recebido pelo sistema SAP da Vodafone, via *web service*. Após validação por parte do sistema da Vodafone, o *stock* de equipamentos deixa de estar no

armazém da Atena T para passar a estar alocado ao cliente. A verificação da validação é feita através do serviço *check consignment status* do *web service* (que é descrito adiante na secção 3.8).

3.3. Modelo de informação

O modelo de informação a ser usado neste projeto utiliza uma tecnologia XML denominada UBL. UBL, ou *Universal Business Language*, é fruto de um esforço internacional com vista a definir um *standard* para documentos empresariais eletrónicos em XML a ser usado, entre outros, nas notas de encomenda e faturas (UBL Docs, 2019). Este modelo foi desenvolvido com recurso a vários elementos *standard* de organizações industriais, pelo que o UBL permite uma ligação transparente com os sistemas existentes nas organizações (Wiki UBL, 2019).

A utilização da UBL associada a *web services* já tem provas dadas em implementações anteriores que a Vodafone PT efetuou. Este projeto irá utilizar a especificação UBL v2.0, mais concretamente dois documentos UBL de topo: *ReceiptAdvice* e *ApplicationResponse*, que se detalham de seguida.

3.4. UBL 2.0 *Receipt Advice*

O modelo UBL do *ReceiptAdvice* é representado na Figura 1. Este documento UBL é um aviso de receção de bens e/ou serviços. Todos os documentos UBL seguem uma hierarquia de elementos que se relacionam entre eles por elementos em comum. Isto é comparável a uma base de dados com as suas tabelas que se relacionam entre si. Neste caso o *receipt advice* é composto pelos seguintes elementos:

- *Document reference* que inclui a data, o tipo de documento e dados identificativos.
- O *Order reference* contém os dados da encomenda do serviço.
- *Customer party* neste é identificado cliente da Vodafone.
- No *Supplier party* vamos identificar a Atena T que é o fornecedor.
- Em *signature* irão constar os dados de validação e métodos de assinatura.

- Em *Receipt line* será criado um destes elementos para cada equipamento entregue ao cliente.
- No *Shipment* vamos encontrar um resumo do que é entregue ao cliente.

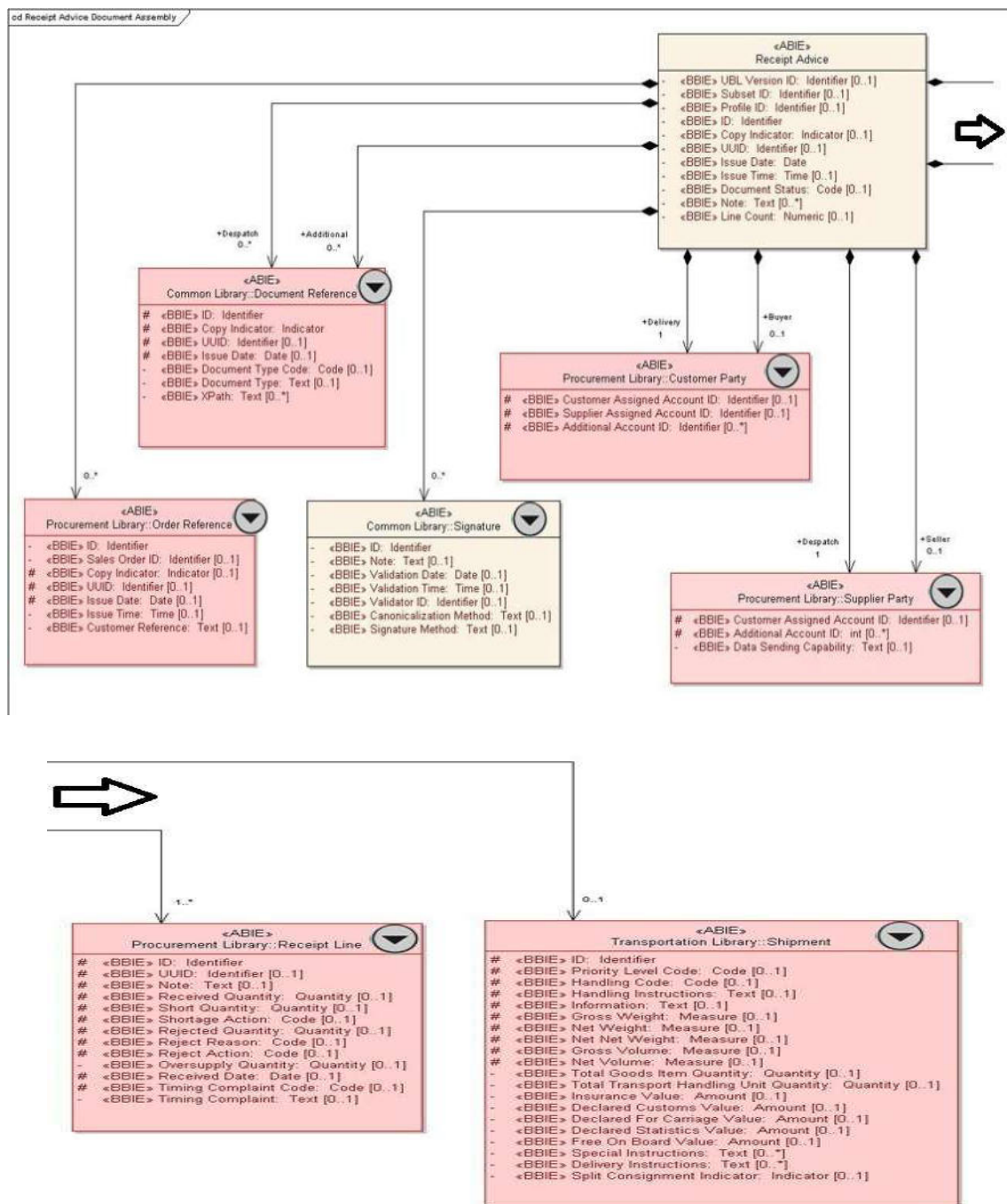


Figura 1. Receipt Advice.

3.5. UBL 2.0 Application Response

O modelo do *Application Response* é representado no diagrama UML da Figura 2. Este documento UBL tem os seguintes elementos:

- Em *Signature* irão constar os dados de validação e métodos de assinatura.
- Em *Party* vão constar os elementos do emissor e do destinatário do documento.
- O *document response* irá conter os elementos de resposta do sistema.

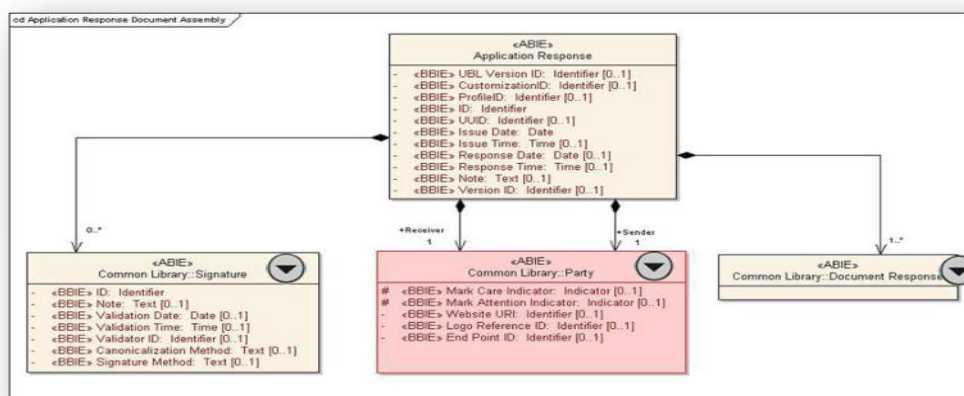


Figura 2 - Application Response

3.6. Componente de ligação

A Vodafone possui um servidor de integração que disponibiliza duas operações, sob a forma de *web service*, para que a Atena T possa enviar os documentos UBL. As operações são as que se descrevem nas secções 3.7 *Consign Stock* e 3.8 *Check Consignment Status*.

Para a operação *Consign Stock*, o sistema de SAP receberá via *web service* da Atena T, a seguinte informação:

- *ADSL Id* ou *Access Id* – identificador do serviço.
- *Account number* – número de conta associado a este serviço
- Identificação do *Field Service* – dados que identificam a Atena T

- Referencias SAP do material - códigos identificativos dos equipamentos instalados.
- Quantidades de material - indica que quantidade foi entregue ao cliente de cada equipamento ou item referente à instalação.
- Números de série dos equipamentos instalados.
- Número de WO (*Workorder*) - corresponde ao nº da ordem de serviço de instalação.

3.7. Consign Stock

A operação *Consign Stock* é utilizada para envio de documento UBL com os dados descritos no ponto anterior, e é feita de forma assíncrona. Na Figura 3 está esquematizada esta operação. O *FieldServices* refere-se ao sistema aplicacional da Atena T que por sua vez interage com o denominado *webmethods* que é a primeira linha do sistema aplicacional da Vodafone. É aqui que encontramos o *web service*. A título ilustrativo são apresentados os movimentos que ocorrem dentro do sistema da Vodafone (*Siebel* e *SAP Legacy*) não vamos interagir diretamente com estes dois elementos.

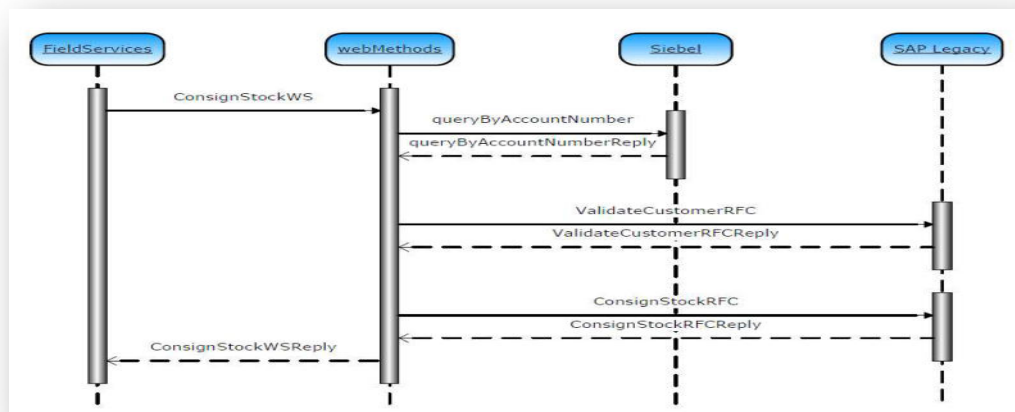


Figura 3. Consign Stock

A Atena T irá invocar o método *ConsignStock* do *web service* e enviar um documento UBL do tipo *ReceiptAdvice* que contém toda a informação necessária para que a consignação do *stock* seja efetuada em SAP. Se algum dos campos obrigatórios não

estiver preenchido ou contenha dados inválidos, o *web service* fecha a ligação de imediato, devolvendo uma informação de erro sob a forma de um documento UBL do tipo *ApplicationResponse*.

Se todos os campos obrigatórios forem válidos, os *webmethods* fazem uma verificação do *account number* e, caso seja válido, então os dados são integrados no sistema de ERP SAP da Vodafone e os equipamentos são alocados ao cliente final. Desta forma deixam de estar sob a alçada da Atena T, sendo assim feita a consignação de *stock*.

3.8. Check Consignment Status

O *Check Consignment Status* tem um método denominado *getConsignStockStatusWS* através do qual é possível obter de forma sequencial o estado de uma WO. Esta operação encontra-se esquematizada na Figura 4.

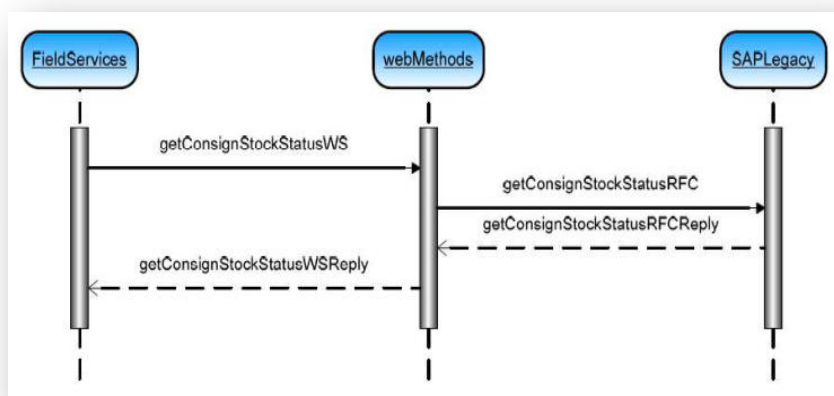


Figura 4. Check Consignment Status

A aplicação da Atena T elabora um *document reference* e invoca o método *getConsignStockStatusWS* dos *webmethods* e então verificado se o número da WO é válido. Em caso negativo, o sistema retorna uma mensagem de erro. Por outro lado, se o número é válido, o sistema interno da Vodafone (*Webmethods* e *SAPLegacy*) irá retornar o estado da integração, pode ser *Pending* ou *Success*. A resposta é feita sob a forma de uma mensagem UBL do tipo *Application Response*.

3.9. Tipo de dados

São utilizados documentos XML com a tecnologia UBL, maioritariamente ReceiptAdvice. Estes são enviados via *web service* para a Vodafone.

4. UNIVERSAL BUSINESS LANGUAGE

4.1. Enquadramento da UBL no projeto

Um dos componentes mais importantes que este projeto vai usar é a *Universal Business Language* (UBL), pois trata-se da base da troca de informação entre as empresas intervenientes nos processos de relacionamentos empresariais. Neste caso, trata-se de uma prestação de serviços da Atena T à Vodafone. Sendo assim, a UBL irá requerer bastante atenção no desenvolvimento deste projeto e terá de ser alvo de um estudo aprofundado. A procura de existência de ferramentas específicas para manipular a UBL, poderá ser fundamental para ajudar a compreender o seu funcionamento e como fazer implementações com ela.

4.2. O que é a UBL?

A UBL é uma linguagem baseada em XML com a finalidade de permitir a troca de documentação comercial entre empresas por meios eletrónicos. Ela é baseada na *standard* da OASIS (UBL Docs, 2019), que consegue responder às necessidades específicas da grande maioria das empresas, é fornecida de forma universal e gratuita, assegura que a informação que uma empresa envia é a suficiente para a empresa destinatária. O esforço desenvolvido para que esta linguagem seja difundida tem como objetivo reduzir a quantidade de papel utilizado nas trocas comerciais. A sociedade está cada mais focada em questões ambientais e a Vodafone é um exemplo ao utilizar a UBL. Adicionalmente, o circuito da documentação empresarial flui mais rapidamente via eletrónica do que pela tradicional via na qual é utilizado o papel. A versão 2.0 da UBL inclui 31 documentos comerciais no formato XML, tais como faturas, notas de crédito, notas de encomenda, catálogos. É uma biblioteca livre e isenta de direitos de autor. A nova versão 2.1 passa a contar com 60 documentos comerciais. A UBL também permite integração direta dos documentos nos sistemas de gestão.

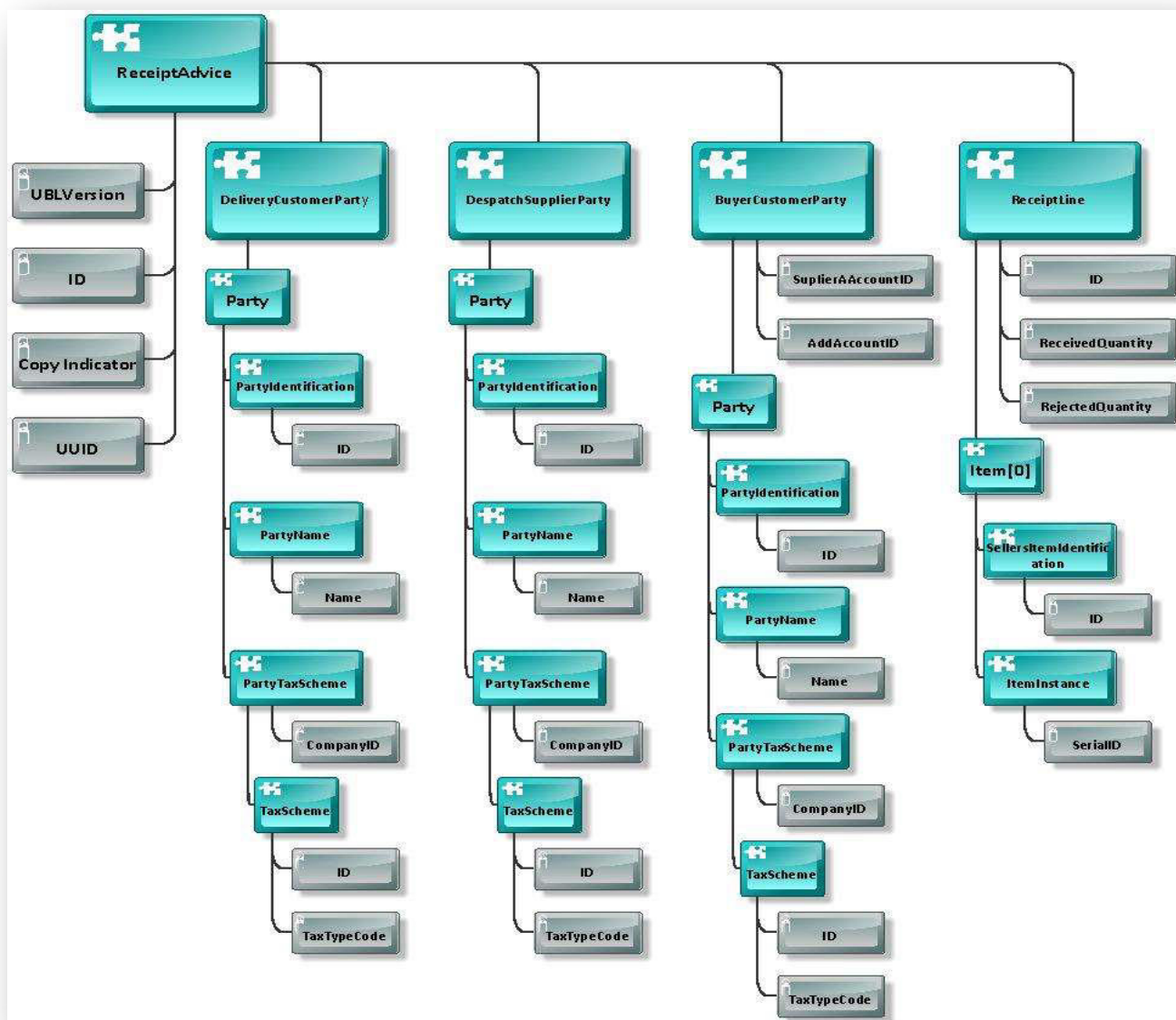
A UBL tem conjuntos de elementos que são utilizados em diversos documentos como:

- *Party*: este elemento identifica uma empresa/entidade. Nele consta o nome, morada, nº de contribuinte, entre outros.
- *Item*: descreve todos os dados dos produtos através de atributos tais como unidades, descrição, nº de série, preço.

A OASIS disponibiliza um conjunto de ficheiros de extensão .xsd, que definem a estrutura de dados dos documentos UBL. Os ficheiros xsd incluem indicadores da existência de dados que são obrigatórios e qual o seu tipo (Ex: *String*, Numérico).

Os ficheiros xsd necessários para este projeto estão referenciados em ficheiro disponibilizado pela Vodafone e que contêm a informação sobre o *web service*. Este será incorporado no projeto de desenvolvimento, utilizando as ferramentas *VisualStudio* e *Netbeans*. Caso seja necessário importar algum xsd adicional, isso poderá ser efetuado através do site da OASIS - <http://docs.oasis-open.org/ubl/cs-UBL-2.0/xsd/common/>.

Os documentos UBL são construídos em forma de árvore. Analisando a estrutura do *receiptAdvice* (Capítulo 3.4), verifica-se que este está na raiz. Seguidamente ramifica-se em entidades *CustomerParty* e *SupplierParty*, em linhas dos produtos *ReceiptLine* e nos extremos temos objetos como a *Item Instance* e *TaxScheme*. O esquema da Figura 5 ajuda a clarificar a estrutura dos documentos UBL, especificamente o *ReceiptAdvice*.

Figura 5 – Estrutura do *ReceiptAdvice*

4.3. Implementação da UBL no projeto

Neste projeto vão ser construídos documentos de acordo com o *standard* da OASIS (UBL) e enviá-los via *web service* para a Vodafone. Serão recebidos pela mesma via documentos UBL que deverão ser tratados na aplicação da Atena T. Estas trocas estarão centradas nos seguintes documentos UBL:

- ✓ *ReceiptAdvice*: de todos os documentos envolvidos neste processo é este o que contém mais informação, como dados das entidades envolvidas, descrição dos

produtos entregues e recolhidos, etc. Genericamente o *ReceiptAdvice* inclui dados dos materiais que foram recebidos e permite efetuar reconciliação destes, é normalmente utilizado pelos departamentos de logística. A Figura 6 contém um exemplo do *ReceiptAdvice* que é utilizado no projeto.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns4:ReceiptAdvice
  xmlns="urn:oasis:names:specification:ubl:schema:xsd:CommonBasicComponents-2"
  xmlns:ns2="urn:oasis:names:specification:ubl:schema:xsd:CommonExtensionComponents-2"
  xmlns:ns3="urn:oasis:names:specification:ubl:schema:xsd:CommonAggregateComponents-2"
  xmlns:ns4="urn:oasis:names:specification:ubl:schema:xsd:ReceiptAdvice-2">
  <UBLVersionID>2.0</UBLVersionID>
  <ID>WorkOrder1</ID>
  <CopyIndicator>false</CopyIndicator>
  <ns3:DeliveryCustomerParty>
    <ns3:Party>
      <ns3:PartyIdentification>
        <ID>337631456</ID>
      </ns3:PartyIdentification>
      <ns3:PartyName>
        <Name>Axentel</Name>
      </ns3:PartyName>
      <ns3:PartyTaxScheme>
        <CompanyID>AXL1</CompanyID>
        <ns3:TaxScheme>
          <ID>504657038</ID>
          <TaxTypeCode>VAT</TaxTypeCode>
        </ns3:TaxScheme>
      </ns3:PartyTaxScheme>
    </ns3:Party>
  </ns3:DeliveryCustomerParty>
  <ns3:DespatchSupplierParty>
    <ns3:Party>
      <ns3:PartyIdentification>
        <ID>supplier party</ID>
      </ns3:PartyIdentification>
      <ns3:PartyName>
        <Name>sup</Name>
      </ns3:PartyName>
      <ns3:PartyTaxScheme>
        <CompanyID>supplier</CompanyID>
        <ns3:TaxScheme>
          <ID>VAT</ID>
          <TaxTypeCode>VAT</TaxTypeCode>
        </ns3:TaxScheme>
      </ns3:PartyTaxScheme>
    </ns3:Party>
  </ns3:DespatchSupplierParty>
</ns4:ReceiptAdvice>
```

```

    </ns3:Party>
  </ns3:DespatchSupplierParty>
  <ns3:BuyerCustomerParty>
    <AdditionalAccountID>buyer3</AdditionalAccountID>
    <ns3:Party>
      <ns3:PartyName>
        <Name>buyer2</Name>
      </ns3:PartyName>
      <ns3:PartyTaxScheme>
        <CompanyID>Buyer1</CompanyID>
        <ns3:TaxScheme>
          <ID>VAT</ID>
          <TaxTypeCode>VAT</TaxTypeCode>
        </ns3:TaxScheme>
      </ns3:PartyTaxScheme>
    </ns3:Party>
  </ns3:BuyerCustomerParty>
  <ns3:ReceiptLine>
    <ID>FS100</ID>
    <ReceivedQuantity>1</ReceivedQuantity>
    <RejectedQuantity>0</RejectedQuantity>
    <ns3:Item>
      <ns3:SellersItemIdentification>
        <ID>520466</ID>
      </ns3:SellersItemIdentification>
      <ns3:ItemInstance>
        <SerialID>CA255105640</SerialID>
      </ns3:ItemInstance>
    </ns3:Item>
  </ns3:ReceiptLine>
  <ns3:ReceiptLine>
    <ID>FS200</ID>
    <ReceivedQuantity>2</ReceivedQuantity>
    <RejectedQuantity>1</RejectedQuantity>
    <ns3:Item>
      <ns3:SellersItemIdentification>
        <ID>500900999</ID>
      </ns3:SellersItemIdentification>
      <ns3:ItemInstance>
        <SerialID>CA255105ppppppp</SerialID>
      </ns3:ItemInstance>
    </ns3:Item>
  </ns3:ReceiptLine>
</ns4:ReceiptAdvice>

```

Figura 6 - Código XML *Receipt Advice*

✓ *Response*: é um documento de resposta que nos indica se o *receiptAdvice* foi rececionado com sucesso ou não, consoante a informação que nele consta. Normalmente tem três elementos: um código de identificação, uma referência da resposta e respetiva descrição (Figura 7).

```
P-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns:Response
xmlns:ns="urn:oasis:names:specification:ubl:schema:xsd:CommonAggregateComponents-2"
xmlns:cbc="urn:oasis:names:specification:ubl:schema:xsd:CommonBasicComponents-2">
      <cbc:ReferenceID>FS0000000099</cbc:ReferenceID>
      <cbc:ResponseCode>0000</cbc:ResponseCode>
      <cbc:Description languageID="en">SUCCESS</cbc:Description>
    </ns:Response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Figura 7 - Código XML Response

✓ *DocumentReference*: Quando se pretende obter informação sobre determinado assunto é usado um *DocumentReference*. Este documento permite obter informação do estado do processamento através do código de identificação (Figura 8).

```
soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:oasis:names:specification:ubl:schema:xsd:CommonAggregateComponents-2"
xmlns:urn1="urn:oasis:names:specification:ubl:schema:xsd:CommonBasicComponents-2">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:DocumentReference>
      <urn1:ID>FS0000000099</urn1:ID>
    </urn:DocumentReference>
  </soapenv:Body>
</soapenv:Envelope>
```



Figura 8 - Código XML *DocumentReference*

✓ *ApplicationResponse*: Este documento inclui a resposta ao *documentreference* e vai informar o estado do processamento e respetivos erros, caso não tenha sido submetido com sucesso (Figura 9).

```

SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns:ApplicationResponse
xmlns:ns="urn:oasis:names:specification:ubl:schema:xsd:ApplicationResponse-2"
xmlns:ext="urn:oasis:names:specification:ubl:schema:xsd:CommonExtensionComponents-2"
xmlns:cbc="urn:oasis:names:specification:ubl:schema:xsd:CommonBasicComponents-2"
xmlns:cac="urn:oasis:names:specification:ubl:schema:xsd:CommonAggregateComponents-2">
      <cbc:ID/>
      <cbc:IssueDate/>
      <cac:SenderParty/>
      <cac:ReceiverParty/>
      <cac:DocumentResponse>
        <cac:Response>
          <cbc:ReferenceID/>
          <cbc:ResponseCode>10001</cbc:ResponseCode>
          <cbc:Description>PROCESSING</cbc:Description>
        </cac:Response>
        <cac:DocumentReference>
          <cbc:ID/>
        </cac:DocumentReference>
      </cac:DocumentResponse>
    </ns:ApplicationResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Figura 9 - Código XML da *Application Response*

A UBL, neste projeto, é utilizada sob a forma de quatro documentos: o *ReceiptAdvice*, o *Response*, o *DocumentReference* e o *ApplicationResponse*. O utilizador

da aplicação introduz todos os elementos das *Workorder* (WO) na base de dados. Após carregar a informação, são elaborados os *ReceiptAdvice* e mostrados em forma de listagem ao utilizador, para que este possa validar antes de enviar. Através da ação do utilizador o *ReceiptAdvice* é enviado via *web service*. Este faz uma validação imediata do documento, ou seja, é verificado se o tipo de documento é o correto, dado que, se for enviado um outro tipo de documento UBL, este é rejeitado. A utilização de documentação UBL torna os processos de troca de informação mais simples, visto que a informação neles contida está sempre no mesmo lugar, podendo ultrapassar facilmente a barreira linguística. Por exemplo, se um funcionário de uma empresa portuguesa que não esteja familiarizado com Mandarim receber um documento de um fornecedor Chinês no formato UBL, será simples identificar a informação e saber o que está em cada campo.

Quando invocado o método *ConsignStock* do *web service*, é enviado um *ReceiptAdvice* e é obtido como resposta um documento UBL do tipo *Response*. Este é tratado pela aplicação da Atena T, de forma a apresentar o resultado ao utilizador, com o qual é possível identificar se o *receiptAdvice* passou a pré-validação do sistema da Vodafone. Caso existam erros, são identificados por mensagem incluída no *Response*. A Figura 10 apresenta o fluxograma desta operação.

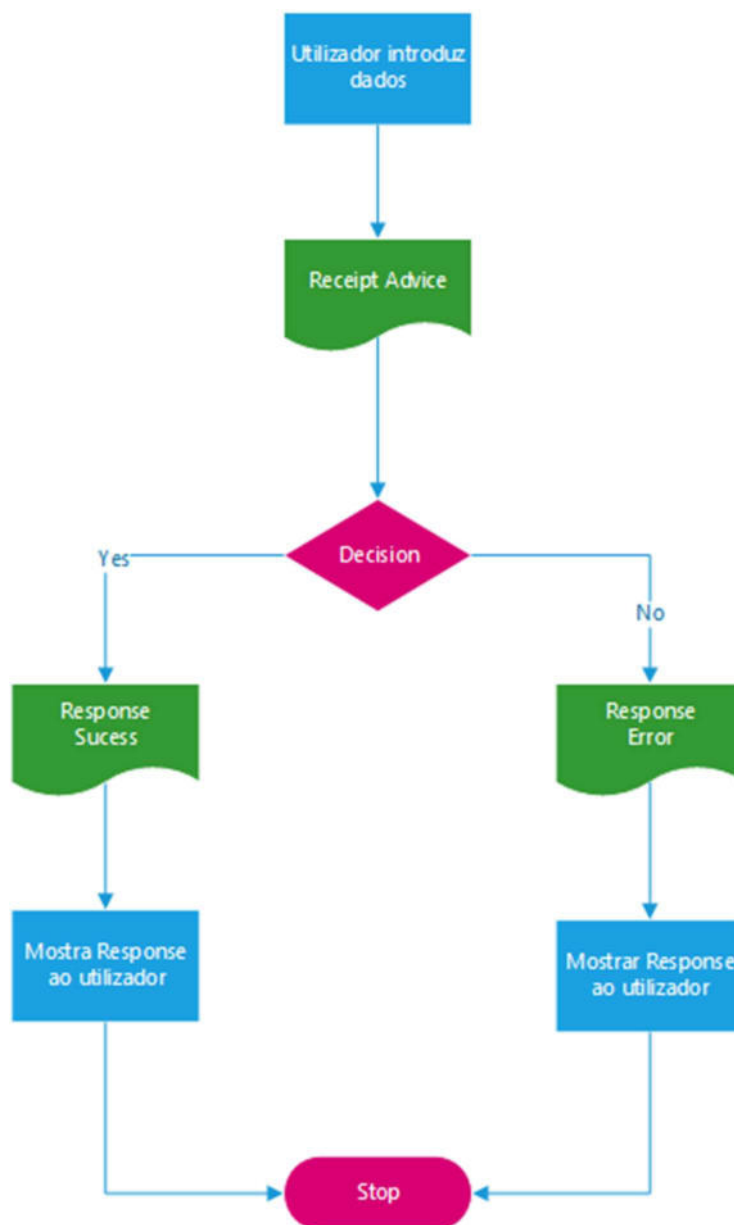


Figura 10 - Fluxograma ConsignStock

O sistema da Vodafone demora cerca de oito a doze horas a integrar e validar os *ReceiptAdvices* que Atena T envia. Passado esse período, torna-se necessário verificar se a validação terminou com sucesso. É então que a aplicação da Atena T emite um novo documento UBL do tipo *DocumentReference*, no qual consta o número da WO, sendo que é com este que se faz a identificação no sistema da Vodafone. Ao ser enviado o *DocumentReference* através do método *CheckConsignmentStatus* do *web service*, de imediato é recebida a resposta do sistema sob a forma de um *ApplicationResponse*. Este inclui informação da validação total do *ReceiptAdvice*, que pode ser de sucesso ou erro.

No caso de erro, é descrita detalhadamente a sua origem, pelo que o utilizador pode identificar as causas e proceder às correções necessárias. O fluxograma da Figura 11 demonstra o circuito do método *CheckConsignmentStatus*.

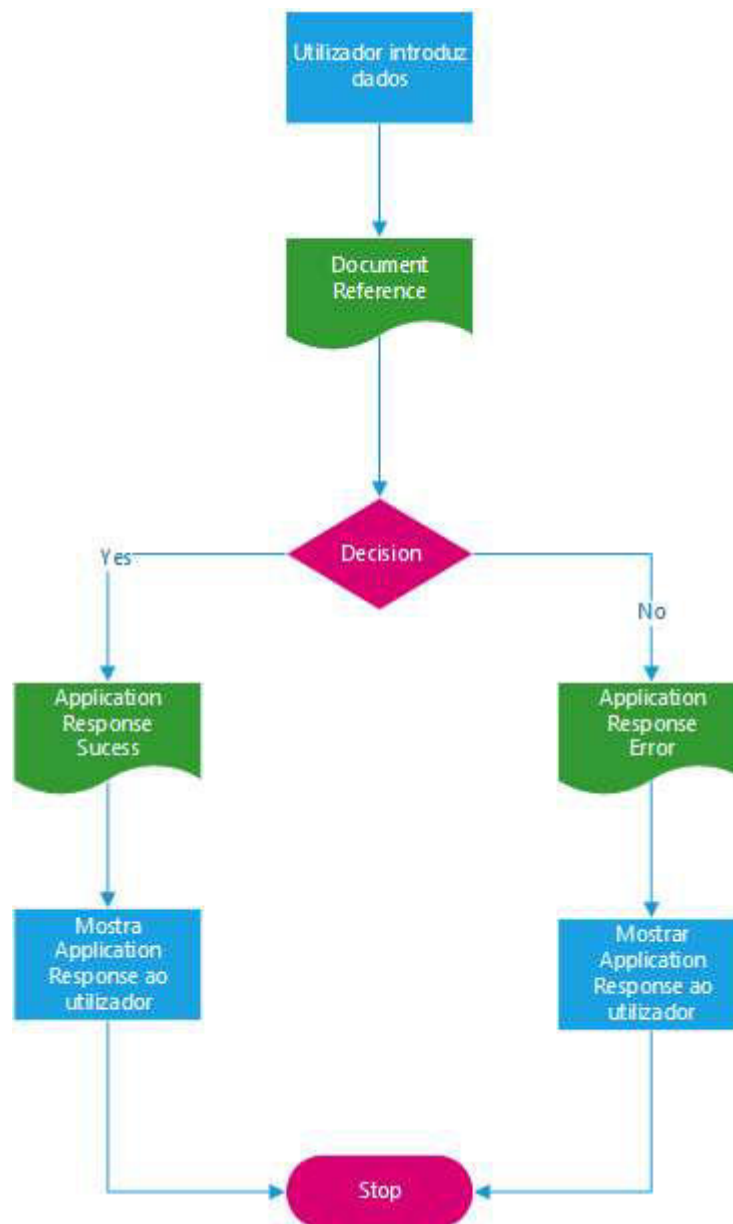


Figura 11 - Fluxograma CheckConsignmentStatus

A informação contida nos *ApplicationResponse* é bastante importante para a gestão da Atena T, logo é necessário desenvolver os métodos necessários para disponibilizar a informação de forma clara para os utilizadores do *backoffice* da Atena T.

Os ficheiros xsd que definem a estrutura de dados dos documentos UBL são designados por *Declared XML Namespaces* (UBL 2.1, 2019). Neste projeto são utilizados os seguintes ficheiros:

- UBL-ApplicationResponse-2.0.xsd
- UBL-CommonAggregateComponents-2.0.xsd
- UBL-CommonBasicComponents-2.0.xsd
- UBL-CommonExtensionComponents-2.0.xsd
- UBL-ExtensionContentDatatype-2.0.xsd
- UBL-QualifiedDatatypes-2.0.xsd
- UBL-ReceiptAdvice-2.0.xsd
- UnqualifiedDataTypeSchemaModule-2.0.xsd
- CodeList_CurrencyCode_ISO_7_04.xsd
- CodeList_MIMEMediaTypeCode_IANA_7_04.xsd
- CodeList_UnitCode_UNECE_7_04.xsd

-

5. DESENVOLVIMENTO DA *DESKCPE* EM JAVA

Este projeto tem como fase inicial o desenvolvimento de uma aplicação direcionada para o *backoffice* da Atena T. A pressão exercida sobre a evolução do trabalho de desenvolvimento aplicativo, pela Atena T, obrigou, nesta fase, a deixar de parte questões estéticas e implementações complexas. As instruções são desenvolver uma aplicação que cumpra apenas o objetivo principal, ou seja, invocar o *web service* e submeter a informação necessária. O prazo previsto de implementação desta fase são seis meses, de maio a novembro de 2013.

A Vodafone aconselhou a utilização de Java neste projeto, tendo em atenção a experiência de implementações idênticas anteriores. Nesta fase, o objetivo principal é desenvolver uma interface que permita invocar o *web service*, por forma a enviar para a Vodafone os *receiptadvice* no formato UBL e obter o *feedback* dos mesmos através da *ApplicationResponse*, também ela no formato UBL. A experiência obtida no âmbito académico centrou-se em J2SE (*Java 2 Standard Edition*), que é a ferramenta de desenvolvimento para a plataforma Java que será a utilizada. A aplicação foi desenvolvida no IDE *Netbeans* na sua Versão 7.3 e é constituída por dois painéis:

- Painel *Consign* que irá interagir com o método *ConsignStock* do *web service*, para elaborar os *ReceiptAdvices* para enviar foi necessário desenvolver um conjunto de classes que os permitem criar. Adicionalmente foi ainda necessário desenvolver as classes para tratamento dos documentos *ApplicationResponse* que nos são enviados pela Vodafone como resposta pelo *web service*.
- Painel *Check Status* é direcionado para o método *CheckConsignmentStatus* do *web service* e foram desenvolvidas as classes para criar o documento UBL do tipo *DocumentReference*. Neste painel também serão tratados documentos *ApplicationResponse*.

Foi ainda usada a ferramenta *Soap UI* para testar o funcionamento do *web service*.

A descrição do *web service* foi disponibilizado pela Vodafone sob a forma de um ficheiro com o formato *Web Service Definition Language* (WSDL). Foi criado um novo projeto no *Netbeans* e o 1º passo foi adicionar o ficheiro WSDL contendo a informação do *web service*, que pode ser importado para o projeto como *Web Service Client*. Este ponto é um dos mais importantes, pois está a ser integrado um elemento externo no projeto. Feita a importação, foram geradas automaticamente e com sucesso várias classes que permitem manipular os objetos e documentos UBL (Figura 12).



Figura 12 – Classes Geradas

Utilizando o código gerado automaticamente chegou a hora de produzir o código para elaborar os *receiptAdvice* necessários para conseguir utilizar o método *ConsignStock* do *web service*. Para tal foram criadas as classes que constam na Figura 13.



Figura 13 - Classes desenvolvidas

O código das classes desenvolvidas encontra-se discriminado no anexo A. Resumidamente cada uma delas tem as seguintes funções:

1. *ApplicationResponseBuild.java* – Trata-se de uma classe de teste que simula um *applicationResponse*, ou seja, a resposta quando o *web service* é invocado. Como nesta fase ainda não existiam condições técnicas para invocar o *web service*, esta classe foi importante porque simula um *ApplicationResponse*, que é o documento que o *web service* envia como resposta.
2. *BuyCustParty.java* – Esta classe possui métodos que permitem criar um objeto do tipo *BuyerCustomerParty*.
3. *ClearReceiptAdvice.java* – O objetivo desta classe é limpar os dados do *ReceiptAdvice* criado anteriormente.
4. *DelCustParty.java* – Classe que permite criar um objeto do tipo *deliveryCustomerParty*. Este inclui elementos da empresa (neste caso, a Antena T) que efetua a entrega/instalação dos equipamentos e serviços no cliente. Os elementos são o número de contribuinte, designação social e códigos identificadores atribuídos pela Vodafone.
5. *DespSupParty.java* – Esta classe contém os métodos necessários à criação do objeto do tipo *DeliverySupplierParty*. Este descreve os elementos como número de contribuinte, designação social e códigos identificadores do fornecedor dos equipamentos ou serviços perante o cliente, ou seja, da Vodafone.
6. *DocReference.java* – Esta classe foi utilizada apenas para testar a criação de um documento UBL do tipo *DocumentReference*.
7. *JframeNoBd.java* – É um *Jframe* que serve de interface gráfica para os utilizadores finais da aplicação. É nesta classe que são chamados a maioria dos métodos constantes nas restantes classes.
8. *RecLineList.java* – Contém métodos para criar objetos do tipo *receiptLine*. Os *receiptLine* são objetos que fazem parte do *receiptAdvice* e descrevem os equipamentos (router, telefone) que são utilizados numa instalação no cliente. São indicadas quantidades, n.ºs de série, códigos de material e códigos de linha do produto.
9. *RespostaBuild.java* – Simula um documento do tipo *ResponseType* que corresponde à resposta do *web service* quando efetuado o *checkStatus*.

10. Classes *RespostaCheckStatus.java* e *RespostaConsign.java* – Manipulam os dados recebidos do *web service* e através dos métodos desenvolvidos o utilizador recebe as respostas sob a forma de texto.

Os métodos que invocam o *web service* têm de ser incluídos no projeto. Esta integração faz-se através um simples “arrastar” da zona do explorador do projeto para a zona de código, conforme se exemplifica na Figura 14.

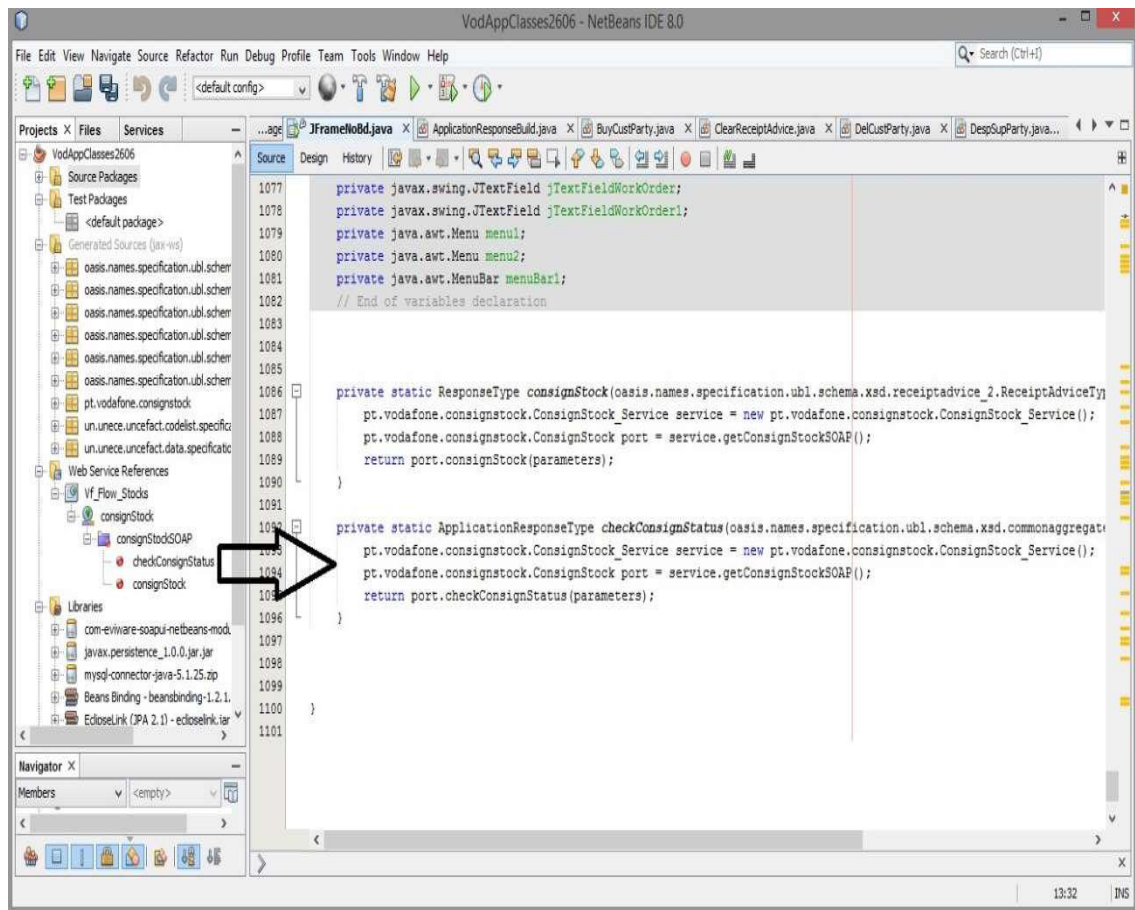


Figura 14 - Integração métodos

Para adicionar os métodos do *web service* basta “arrastar” os pontos vermelhos para a classe que pretendemos e o código é gerado automaticamente.

Inicialmente foi definido que existiria um gestor no projeto que tinha a seu cargo a imagem, e que elaboraria o *design* de todos os elementos que fazem parte do projeto total, incluindo o *design* gráfico desta aplicação. Nesta fase, como o elemento imagem ainda não tinha sido disponibilizado, avançou-se para o desenvolvimento do código e de uma *interface* simples, aguardando futuras orientações em relação ao *design*.

A interface que permite ao utilizador final interagir com o *web service* está representada na Figura 15 e na Figura 16.

The screenshot shows a Java Swing window titled 'Menu Edit'. It contains two main sections: 'Workorder' and 'Resposta'.

Workorder Section:

- Fields: Workorder nº, Data (with example 'ex: 2013-10-22'), FS ID, UUID, Conta Cliente, ADSL Id, Contribuinte nº, and Nome Cliente.
- Table with 5 columns: Item ID, Received, Rejected, Material, and Nº série. It has 4 empty rows.
- Buttons: 'Enviar' and a large empty *JTextArea* on the right.

Resposta Section:

- Fields: Ref Id, Código, and Descrição.

Figura 15 – Painel *Consign*

Como se pode ver na Figura 15, a interface é constituída por duas áreas. Uma refere-se à consignação/envio dos *receiptAdvice* e intitula-se *workorder*. Aqui são introduzidos os dados referentes a uma instalação do serviço ADSL Vodafone no cliente, como a identificação do cliente e equipamentos colocados à sua disposição. Ao acionar o botão enviar vamos visualizar no elemento à esquerda do tipo *JTextArea*, o conteúdo XML do *receiptAdvice*.

A outra área é designada por “Resposta” aqui será visualizado a resposta do *web service* referente ao *receiptAdvice* que foi enviado.

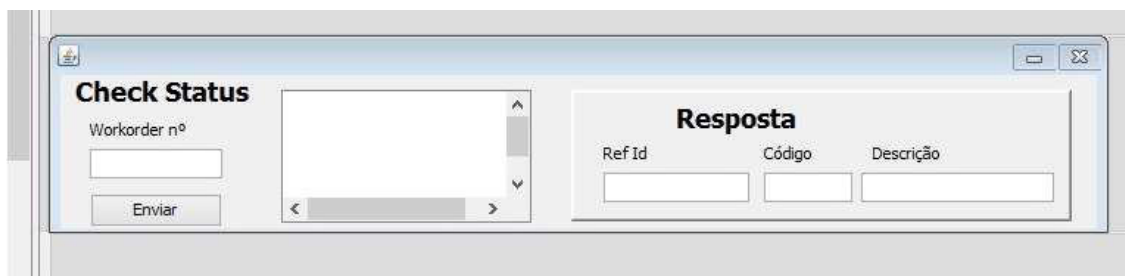


Figura 16 - Painel *check status*

O painel de *CheckStatus* (Figura 16) irá permitir verificar o estado das *workorders* submetidas, ou seja, permite confirmar se a informação que consta no respetivo *receiptAdvice* enviado foi aceite no ERP SAP da Vodafone, com sucesso ou se existem erros para serem tratados.

5.1. Segurança

A segurança das comunicações entre a Atena T e a Vodafone é feita através de IPSec (Oracle, Docs Oracle, 2019). O *web service* encontra-se alojado nos sistemas da Vodafone e para ser acedido houve necessidade de subscrever um serviço de IP fixo. Este IP é reconhecido pelo sistema da Vodafone através de um túnel de IPSec, que fornece autenticação ao nível da rede, verifica a integridade dos dados e comunicação utilizando criptografia e chaves de 128 Bits. Caso outro IP tente aceder ao *web service*, a ligação é logo rejeitada. O IPSec suporta-se em três protocolos:

- *Authentication Header* (AH) – Este garante a autenticidade e integridade dos dados contidos num pacote. É utilizado quando não existe necessidade de confidencialidade da informação.
- *Encapsulating Security Payload* (ESP) – É o mais utilizado pois além da autenticação dos pacotes encripta a informação neles contida.
- *Internet Key Exchange* (IKE) – Responsável pelas ligações ESP ou AH, após verificação de elementos como chaves e algoritmos.

Assim, fica assegurada a integridade dos dados, controlo de acessos e privacidade de dados. Houve necessidade de contratar um acesso em fibra, dedicado, com

IP fixo e instalar um *router* Cisco 891 fornecido pela Vodafone. Além do túnel IPSec, existe um sistema de autenticação, sem o qual a ligação é cancelada.

De acordo com informação disponibilizada inicialmente pela Vodafone, o protocolo de comunicação estava baseado em HTTP (w3schools, 2019). No entanto, chegada a fase de testar a comunicação da *deskCPE* com o *web service*, verificou-se que a comunicação tinha evoluído a nível de segurança, passando a utilizar o protocolo HTTPS (Globalsign, 2019), que permite a encriptação dos dados que são enviados. Esta alteração de protocolo obrigou que fossem feitas algumas alterações no código, nomeadamente através da inclusão de *Authenticators* (Oracle, Class Authenticator, 2019), *PasswordAuthentication* (Oracle, Class PasswordAuthentication, 2019) e *X509TrustManager* (Oracle, Class X509TM, 2019). Os objetos da classe *authenticator* em Java fazem autenticação numa rede. Através dos métodos desta classe podemos efetuar a autenticação através de *username* e *password*, em que estes dois elementos ou credenciais constituem um *PasswordAuthentication*. Quando o *web service* é invocado com o protocolo HTTPS, é exigida a utilização de um certificado, daí a utilização da classe *X509TrustManager*.

5.2. Testes

A fase de testes tem por objetivo verificar se a aplicação consegue interagir com sucesso com o *web service*. Os testes foram definidos pela Vodafone e quando a Atena T considerou que a aplicação se encontrava preparada para ser testada foi iniciada esta fase. O documento disponibilizado pela Vodafone encontra-se reproduzido como ANEXO B – TESTES DA APLICAÇÃO.

A Vodafone disponibilizou dados de teste para serem enviados pela aplicação da Atena T e, se os dados chegarem corretamente ao sistema da Vodafone, então os testes terão sido executados com sucesso.

Os testes da aplicação dividem-se em duas fases. A primeira inclui cinco conjuntos de testes que são efetuados invocando o endereço de teste do *web service*:

- Teste 1, de consignação – serve para indicar os equipamentos que foram aplicados numa instalação de serviço ADSL. Este teste inclui 3 cenários: o 1.1 que utiliza apenas equipamentos com número de série, e o 1.2 que só utiliza materiais sem número de série como comando

dos equipamentos e o último (1.3) que são utilizados equipamentos com e sem número de série. No final de cada conjunto de testes, é necessário indicar o resultado e aguardar validação por parte da Vodafone de modo a avançar para teste seguinte ou repetir.

- Teste 2, de desconsignação – será testada a devolução de equipamentos que podem resultar, por exemplo, de um processo de desinstalação de serviço ADSL por opção do cliente. Existem os seguintes 6 cenários neste teste:
 - 2.1 - Desconsignação total só com números de série. São utilizados dados de vários equipamentos com números de série, uma TV *box* que permite visualizar canais de TV e dois *routers Technicolor TG 784n*.
 - 2.2 - Desconsignação parcial só com números de série. É feito um teste de desconsignação com 2 equipamentos apenas.
 - 2.3 - Desconsignação total só com quantidades: os tipos de equipamentos utilizados neste teste não possuem número de serie, como comando de TV *box*.
 - 2.4 - Desconsignação parcial só com quantidades. Este cenário é semelhante ao anterior, diferem os dados do cliente.
 - 2.5 - Desconsignação total com números de série e quantidades. Neste cenário são utilizados quatro equipamentos que podem possuir ou não número de série.
 - 2.6 - Desconsignação parcial com números de série e quantidades. Este processo é semelhante ao descrito no ponto anterior, mas com apenas dois equipamentos.
- Teste 3: Troca de material – neste teste foram utilizados dois cenários de troca de materiais.
 - 3.1 Troca parcial de material. É feita uma troca parcial de um equipamento do cliente, neste caso é uma *Box* de IPTV que tem número de série.
 - 3.2 Troca total de material. Neste cenário é feita a troca de todos os equipamentos que se encontram instalados no cliente.

- Teste 4: Criação de processo *end-to-end* com o mesmo número de série. Neste vão ser utilizados alguns dos cenários dos testes 1, 2 e 3 para se efetuar um único teste que permitirá verificar a fiabilidade da aplicação.
- Teste 5: Movimentos de material entre empresas de *Field Service* – este teste é composto por 9 cenários.
 - 5.1 Desconsignação de material na totalidade com centro 8300: cada empresa fornecedora de serviços (*Field Service*) da Vodafone pode ter vários centros de operações (Ex: Lisboa e Porto). Neste cenário é feita uma desconsignação de material de um centro.
 - 5.2 Desconsignação de material na totalidade com centro 8500. Este cenário é idêntico ao descrito em 5.1 sendo que o centro é diferente.
 - 5.3 Desconsignação de material na totalidade com centro 8600. Teste semelhante aos dois anteriores.
 - Os cenários 5.4, 5.5 e 5.6 são de desconsignação de material parcial, ou seja, em cada um deles é feita uma desconsignação parcial de um ou dois produtos. O centro é sempre diferente. Os cenários 5.7, 5.8 e 5.9. são idênticos aos anteriores, no entanto é efetuada uma troca de material, ou seja, uma recolha e uma entrega.

O primeiro teste da fase 1 foram reportados apenas erros relacionados com o preenchimento dos dados no formulário que a seguir são descritos:

- A colocação do *ItemId* no Painel *Consignstock*, o documento dos testes não refere que esta coluna deve ser preenchida com FS0000001 na linha 1, e nas seguintes linhas incrementar 1 unidade (FS0000000X)
- No *Work Order* ID não deve constar o prefixo WO – deve ser só preenchido o nº, por ex. 4000001
- Não se pode enviar as tags “*ReceivedQuantity*” e “*RejectedQuantity*” ao mesmo tempo. Quando for uma consignação deve surgir apenas a

tag “ReceivedQuantity” e quando for uma desconsignação deverá ser enviada a *tag “RejectedQuantity”*

- Nos materiais seriados é para enviar uma quantidade por cada linha. ou seja, no teste 1.1 surge o material 108003901 com 2 quantidades e 2 n°s de série. Não se pode enviar *ReceivedQuantity* = 2 e *SerialID* = 0876FF8688E1 + 0876FF87457D; tem de se enviar 2 *receipt lines* com 1 quantidade e 1 n° série em cada uma e o mesmo material

A introdução de dados passou a ser efetuada de acordo com o solicitado pela Vodafone. Esta primeira fase de teste iniciou-se em outubro de 2013 e finalizou com sucesso em novembro de 2013.

A segunda fase é composta por seis testes. A estrutura destes é idêntica à primeira fase e é efetuada invocando o endereço de produção, ou seja, para uma interação com o ambiente real.

Esta fase é constituída pelos seguintes testes:

- Teste 1: Consignação com números de série e quantidades, utilizando apenas um cenário.
- Teste 2: Desconsignação com números de série e quantidades, utilizando apenas um cenário.
- Teste 3: Troca de material: é efetuado um cenário de troca de material a um cliente.
- Teste 4: Criação de processo *end-to-end* com o mesmo número de série. São efetuados 4 passos: consignação, desconsignação, consignação a outro cliente e trocar ao cliente.
- Teste 5: Movimentos de material entre empresas de *field service*. São utilizados os cenários 5.1 a 5.8, que são idênticos aos usados na primeira fase de testes.
- Teste 6, composto pelos seguintes cenários:
 - 6.1 Consignação de material TELF – Novo Cliente. Neste cenário é consignado ou instalado um telefone num novo cliente.

- 6.2 Consignação com números de série e quantidades ADSL e TELF. São consignados 2 equipamentos com número de série.
- 6.3 Desconsignação com materiais ADSL e TELF. É feita a desconsignação de 2 equipamentos seriados referentes a um processo de retirada de equipamentos que estavam instalados no cliente.
- 6.4 Movimentos de material entre empresas de *field service*/desconsignação com materiais ADSL e TELF outro centro. Este cenário é idêntico ao anterior.

Nesta fase foi apenas reportado que no *ckeckConsignStock* o ID a enviar não é a WO, mas sim o UUID. A 2ª fase de testes terminou em 18 de dezembro de 2013 e após validação por parte da Vodafone, a aplicação estava preparada para a fase de produção.

5.3. Distribuição

Durante o processo de desenvolvimento aplicativo o *netbeans* permite que o código seja executado diretamente, no entanto para que as aplicações sejam executadas fora do ambiente do *netbeans* é necessário produzir um ficheiro do tipo *.jar* (Oracle, 2020), este ficheiro agrega os diversos componentes de um projeto (Classes, imagens e outros), neste caso, desenvolvido em *netbeans*.

O ficheiro *jar* é criado no *netbeans* através dos seguintes passos:

1. Clicar no botão direito do rato em cima do nome do projeto.
2. Selecionar propriedades.
3. Clicar em Packaging.
4. Ativar opção Build JAR after Compiling.
5. Ativar opção Compress JAR File.
6. Clicar OK para aceitar as alterações.
7. Clicar no botão direito do rato em cima do nome do projeto
8. Selecionar Build ou Clean and Build.

O ficheiro jar obtido não pode ser executado diretamente no ambiente Windows para isto ser possível é necessário transformá-lo num ficheiro executável denominado *Windows native executables* ou exe.

Os testes determinam que sejam produzidas duas versões do *deskCPE*:

- A primeira com o endereço de testes, para ser apenas utilizada na primeira fase de testes, faz parte do desenvolvimento e implementação do *deskCPE*, mas também se destina a ser utilizada pelo *backoffice* da Atena T. Esta fase permitiu verificar se os utilizadores do backoffice conseguem interagir de forma simples e se consideram o ambiente do *deskCPE user-friendly*, ou seja, de fácil utilização e visualmente agradável.
- A segunda versão com o endereço de produção, que foi usada na segunda fase de testes destina-se a ser utilizada pelo *backoffice*, que após finalização dos testes com sucesso passou a ser a ferramenta de utilização no processo de consignação e desconsignação de materiais utilizados no processo de gestão do cliente Vodafone.

Durante o desenvolvimento e implementação do *deskCPE* este é executado através do IDE Netbeans, no entanto não é possível e não faz sentido implementar o *deskCPE* desta forma para os utilizadores do *backoffice*. Sendo assim ambas as versões do *deskCPE* para serem utilizadas pelo *backoffice* passaram por um processo de transformação, o qual consiste em passar um ficheiro .jar num executável do tipo .exe.

Esta transformação é obtida através do *software Launch4J*.

O *Launch4J* é muito simples de utilizar. Selecionando o separador "*Basic*", coloca-se o caminho onde o executável será guardado em "*Output file*". Em "*Jar*" seleciona-se o arquivo .jar que pretendemos converter para .exe. No separador "*JRE*" é necessário indicar a versão mínima de JRE que será exigida para executar a aplicação em "*Min JRE version*". No separador "*Version Info*" podemos colocar informações do executável. É ainda possível definir o ícone para o executável no separador *Basic*. A Figura 17 - **Software Launch4J** mostra o separador *basic* do *software*.

Depois de tudo configurado, aciona-se o botão de *Build wrapper* e a nossa aplicação é construída. Nesta fase o *Launch4J* irá pedir para indicar uma localização para guardar o ficheiro de configuração.

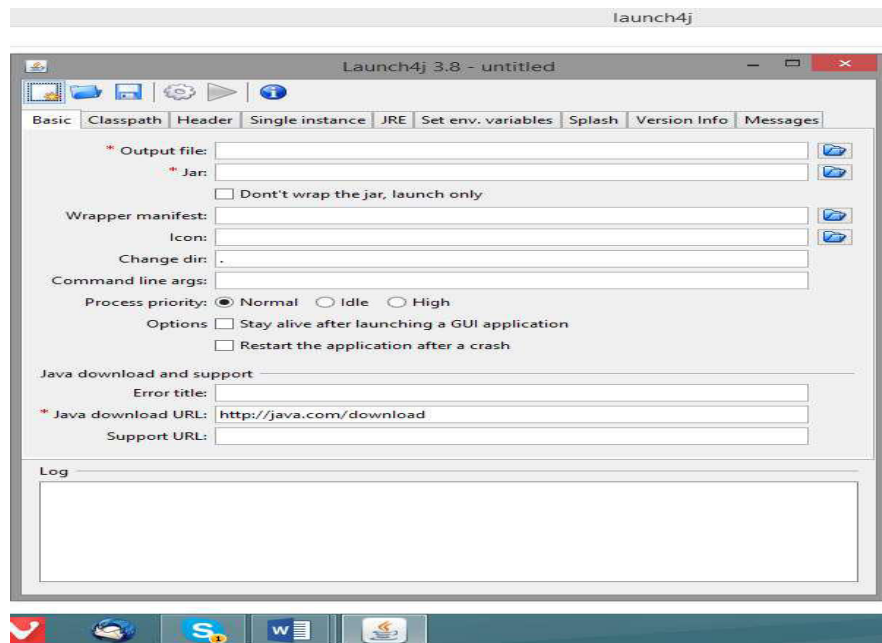


Figura 17 - Software Launch4J

Com o *deskCPE* pronto chega o momento de parametrizar o ERP PHC da Atena T para que este esteja preparado para gerir este novo negócio com o cliente Vodafone.

6. PARAMETRIZAÇÃO DO ERP PHC

O PHC é um sistema de ERP que combina vários módulos interligados, o que permite obter informações de gestão bastante fiáveis. O PHC é apenas suportado por ambientes *Windows*, e as suas bases de dados podem ser geridas apenas pelo *SQL Server* em qualquer uma das suas versões incluindo o *Express*. Tem uma potente *Framework* que permite aos utilizadores avançados efetuar várias personalizações. As linguagens de programação utilizadas são o *X-Base* e o *Fox Pro*.

Esta fase tem como objetivo preparar o PHC para que fique moldado de forma a permitir gerir esta nova área de atividade com o cliente Vodafone. Sendo assim, foi necessário adaptar os documentos utilizados atualmente como a folha de obra, documento de entrada em armazém, guia de transferência entre armazéns. Nestes documentos em particular, é necessário adicionar os novos campos essenciais para dar resposta adequada ao cliente. Houve também necessidade de implementar no sistema a opção de tratamento de n.ºs série, pois esta funcionalidade não estava ativa.

Os documentos que têm de ser adaptados são denominados no PHC por “Dossiers internos”, neste âmbito por defeito já vêm criados um conjunto de dossiers que servem genericamente para a maioria das empresas, estes são, a título de exemplo, Nota de encomenda, orçamento e proposta. No entanto é possível criar tantos quantos dossiers a empresa necessite, neste caso da Atena T são então criados, especificamente para este projeto, os dossiers folha de obra, documento de entrada em armazém e guia de transferência entre armazéns. A Figura 18 mostra o menu do dossier interno.

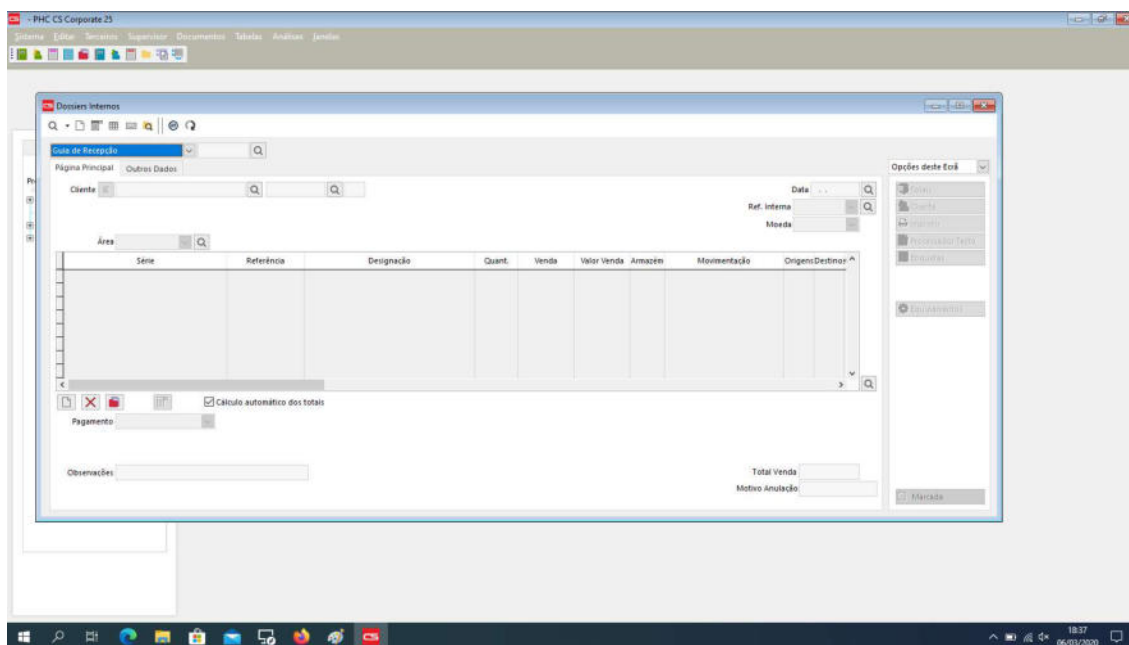
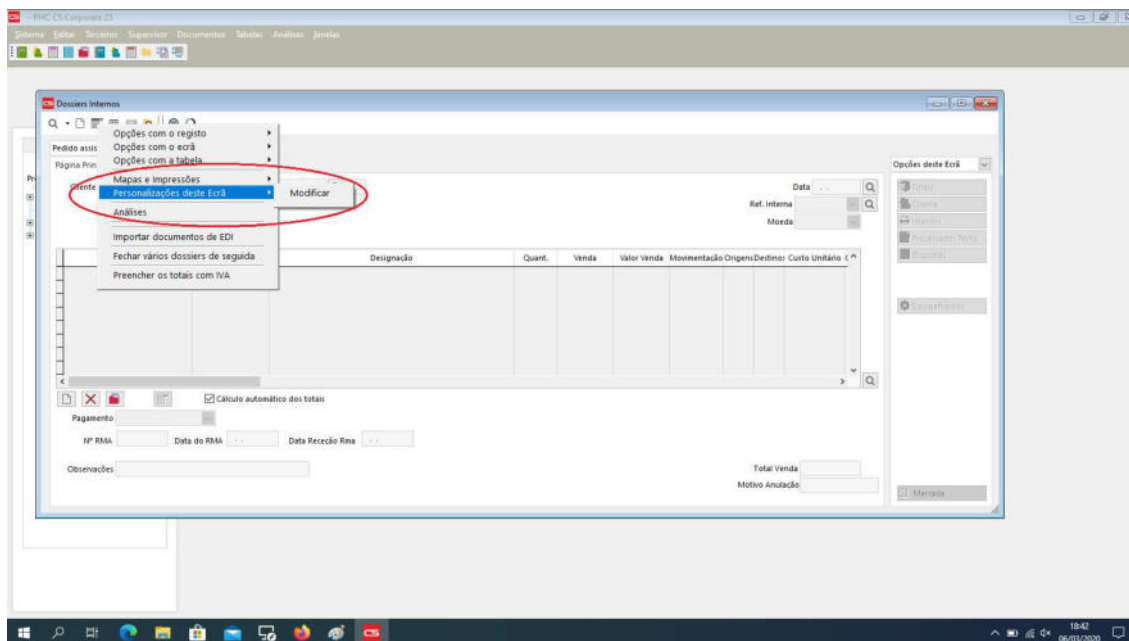


Figura 18 - PHC Dossier Interno

Adicionalmente o menu do dossier interno permite aceder a parametrizações que permitem adicionar campos ao dossier interno conforme Figura 19.

Figura 19 - PHC Acesso parametrização dossier



Ao aceder à parametrização é possível adicionar nos objetos como campos disponíveis (conforme indicado na Figura 20), imagens, formas, texto e outros.

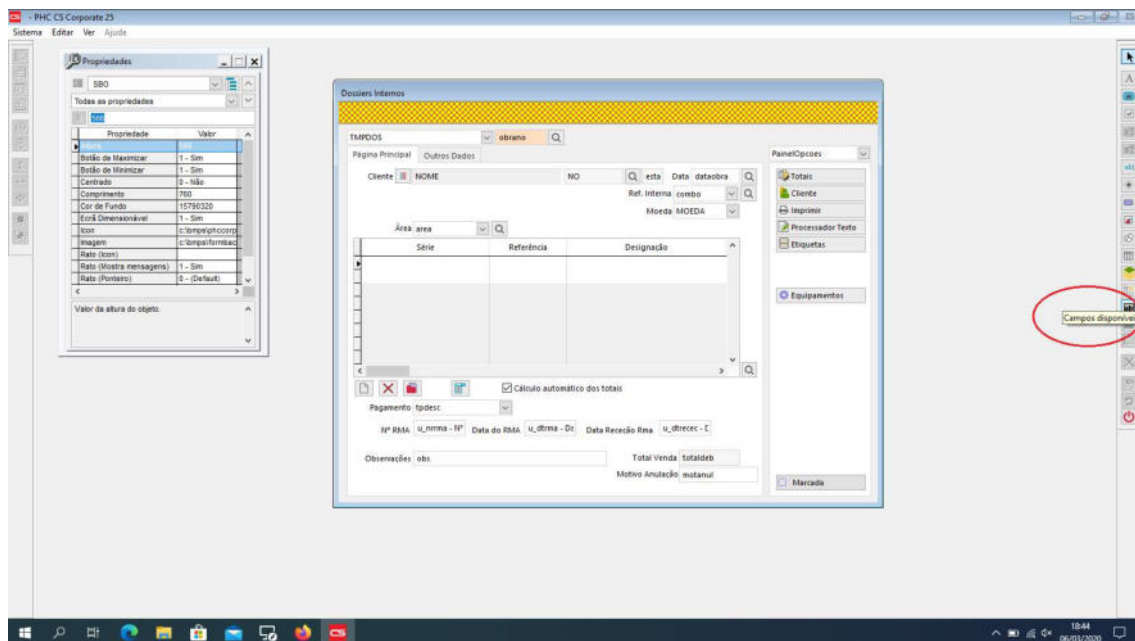


Figura 20 - PHC Menu de parametrização dossier

É ainda possível criar campos de utilizador que permitem que o utilizador não fique limitado ao que o PHC disponibiliza por base, ou seja, neste caso há a necessidade de existir um campo no dossier para a WO e de base o PHC não possui um campo específico para WO então é possível criar um campo de utilizador, esta funcionalidade faz parte da *framework* do PHC, a Figura 21 mostra como se acede.

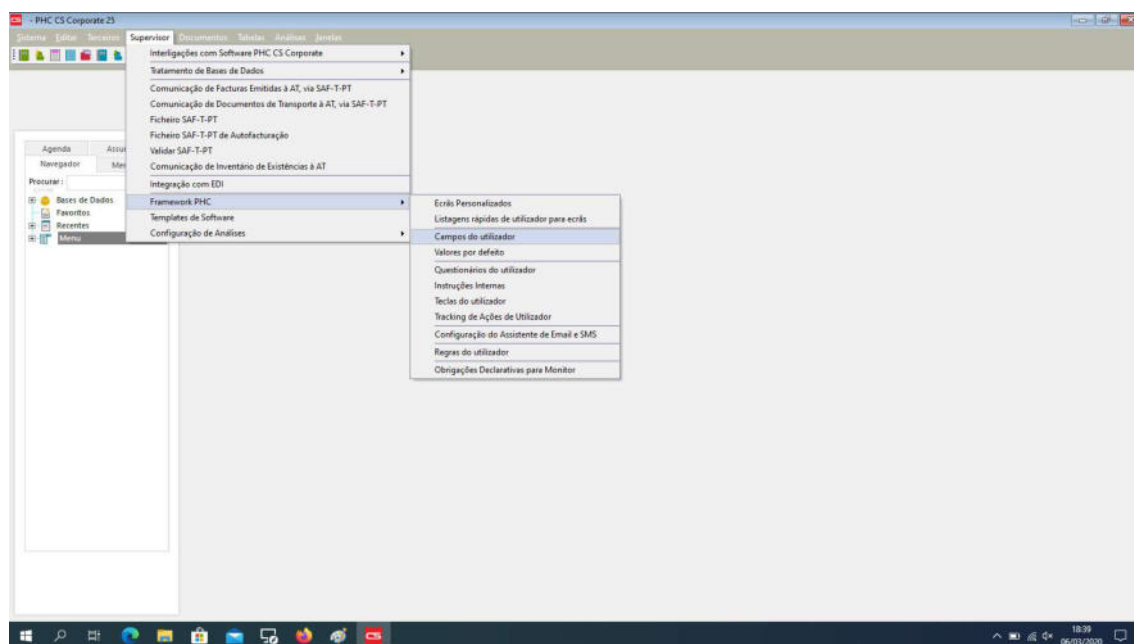


Figura 21 - PHC Acesso Campos de utilizador

Os campos de utilizador permitem definir regras na sua criação como:

- Tipo de dados: Numérico, caracteres.
- Nº de caracteres e/ou dígitos permitidos.
- Decimais – no caso de numérico quantas casas decimais tem.
- Associar uma imagem ao campo.
- Adicionar um comando em Transact SQL.

A Figura 22 mostra o formato do painel de configuração dos campos de utilizador.

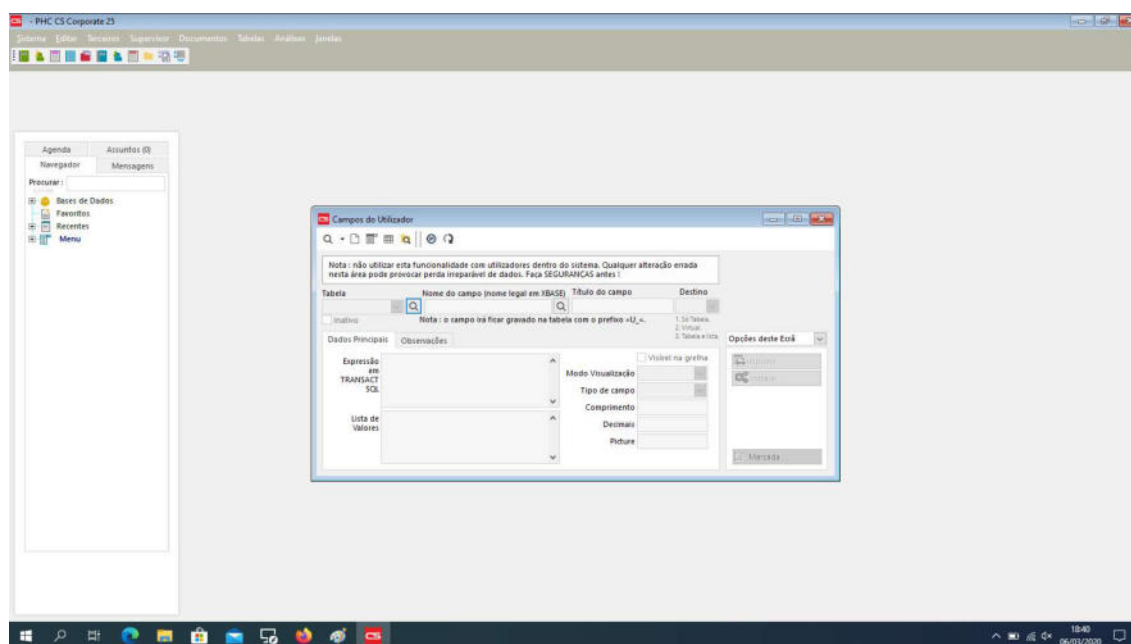


Figura 22 – PHC Painel Campos de utilizador

Para estes dossiers também houve necessidades de ativar a utilização de números de série dos equipamentos, que é efetuada conforme demonstra a Figura 23.

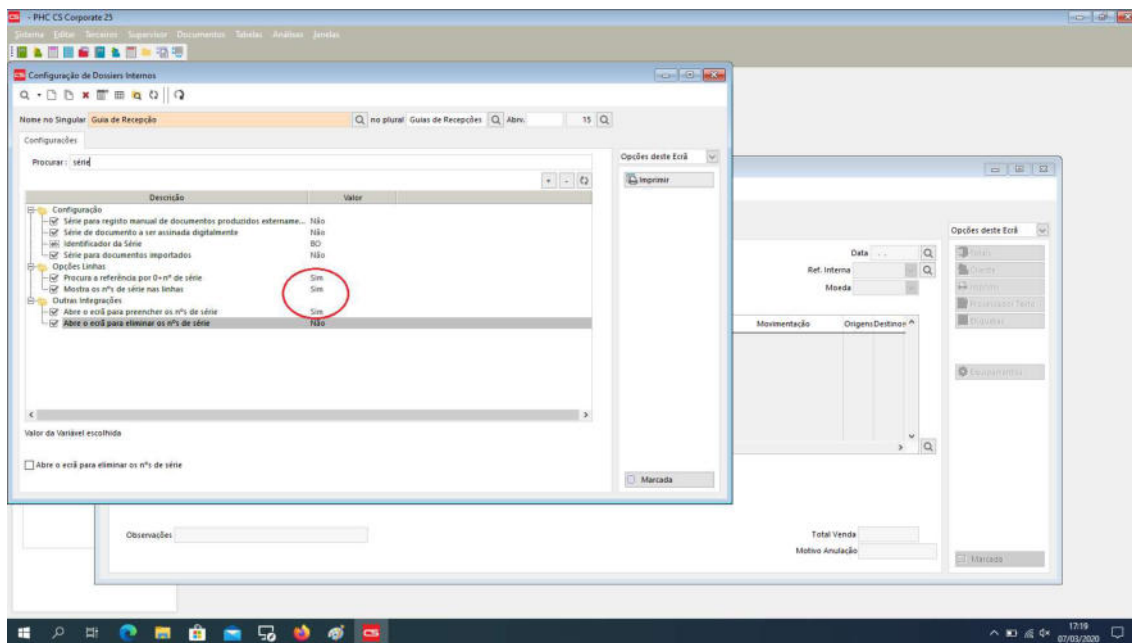


Figura 23 - PHC Configuração nº série no Dossier

No sistema PHC é no menu Stock e Serviços que são criados novos produtos ou serviços, como foi referido anteriormente serão utilizados equipamentos com nº de série logo é necessário ativar na ficha de produto se o mesmo tem nº de série, conforme indicado na Figura 24, esta opção tem de ser tomada quando o produto é criado pois mais tarde o sistema PHC já não deixa alterar.

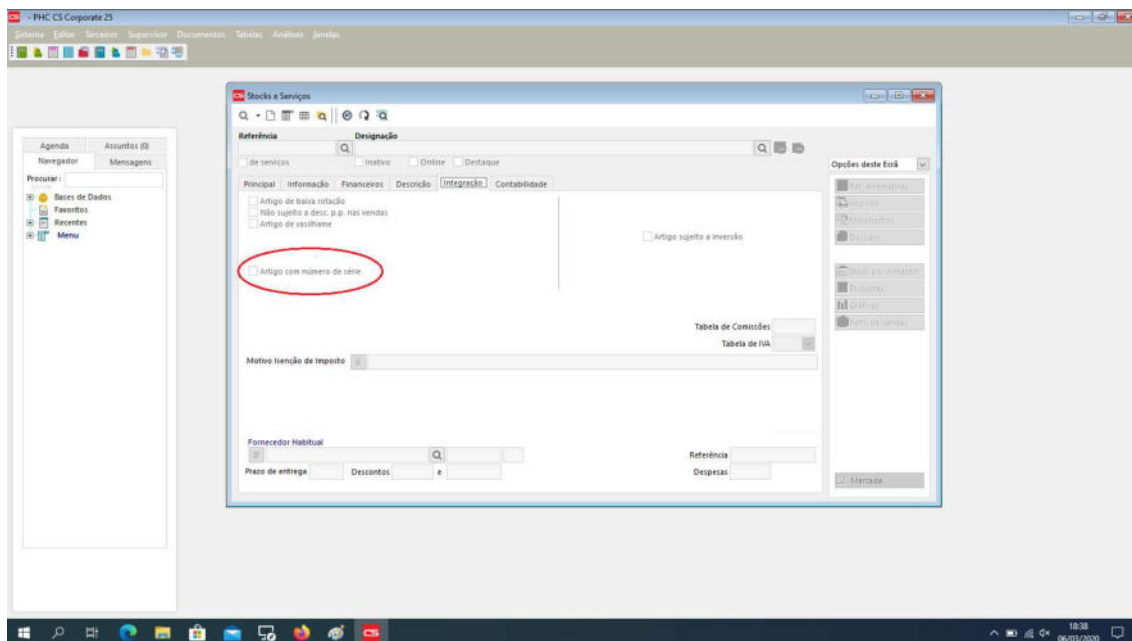


Figura 24 - PHC Ativar nº Série em Stock e serviços

Criados os produtos (Equipamentos) seriados, sempre que é adicionado um novo dossier interno, se o produto obriga a introdução de nº de série o sistema alerta o utilizador através uma janela conforme a Figura 25.

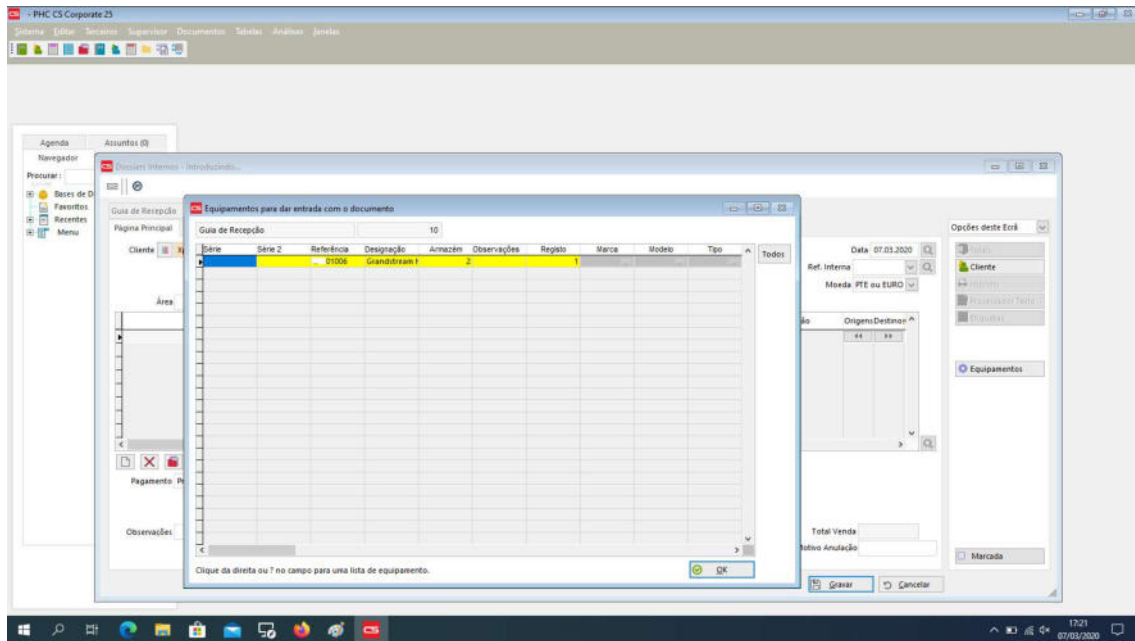


Figura 25 - PHC Janela introdução nº série

Com a finalização da parametrização do ERP e altura de avançar para o desenvolvimento do *webCPE*.

7. DESENVOLVIMENTO DO WEBCPE

O projeto entra numa nova fase de desenvolvimento, após a colocação em produção do *deskCPE*, é altura de avançar com o *webCPE* que é uma plataforma direcionada para ambiente *web* que pode ser acedida em qualquer local sem estar dependente de uma instalação de um executável.

Iniciada esta nova etapa do projeto do desenvolvimento do *webCPE* que será disponibilizado via *browser* (*Internet explorer*, *chrome*, *Firefox* e outros) com ligação a um sistema de base de dados e um serviço responsivo, acessível via *smartphone*. É nesta fase que a *deskCPE* vai evoluir, ou seja o *deskCPE* passa para segundo plano para dar lugar ao *webCPE* que será seguramente melhor pois passa a ser acedida através de *browser*, o utilizador do *backoffice* não terá de se preocupar se tem o *deskCPE* para trabalhar, com o *webCPE* basta abrir o *browser* de internet a aceder ao endereço do *webCPE*. Adicionalmente o *webCPE* já terá autenticação de utilizador, serão definidos níveis de acesso, passa a existir um foco no aspeto segurança, e as preocupações com o *design* já estarão bem presentes.

7.1. Ferramentas de desenvolvimento

O *Visual Studio* 2012 e 2013 foram utilizados no desenvolvimento do *webCPE*. A opção pela utilização da versão mais atual do *Visual Studio*, a de 2013, revelou-se errada pois não inclui a funcionalidade do *Web Configuration* (Microsoft, 2019), sendo assim o projeto continuou a ser desenvolvido com o *Visual Studio* 2012. Foi utilizado o *Sql Server 2014 Developer Edition* para implementação e gestão da base de dados.

7.2. Estrutura do *webCPE*

O *webCPE* é constituído por duas áreas, uma pública e outra reservada.

7.2.1. A área pública

A área publica do website tem uma estrutura tradicional composta pelas páginas *Home*, *Sobre* e *Contactos*.

7.2.1.1. Home

Esta é a página de entrada do *website* principal e aqui podemos aceder aos diversos perfis de acesso da área reservada (Administração, *Backoffice* e Técnicos). Também estão presentes os habituais links de navegação para as restantes páginas do *website* (Figura 26).

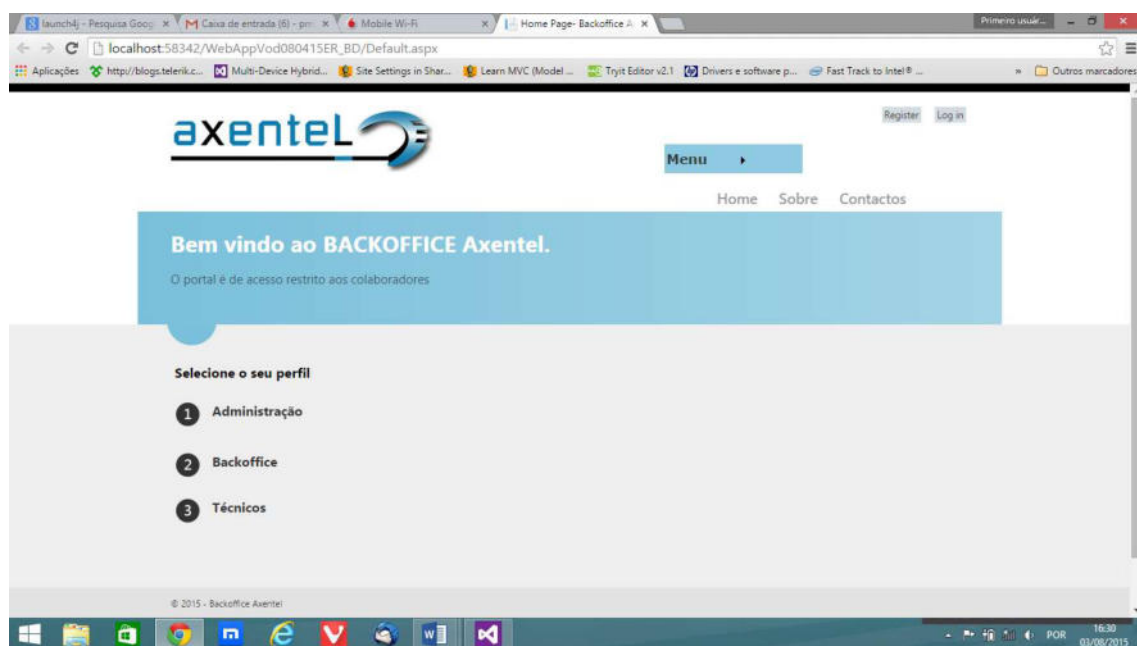


Figura 26 - Página *Home*

7.2.1.2. Contactos:

Esta página informa quais são os contactos gerais da empresa (Figura 27).

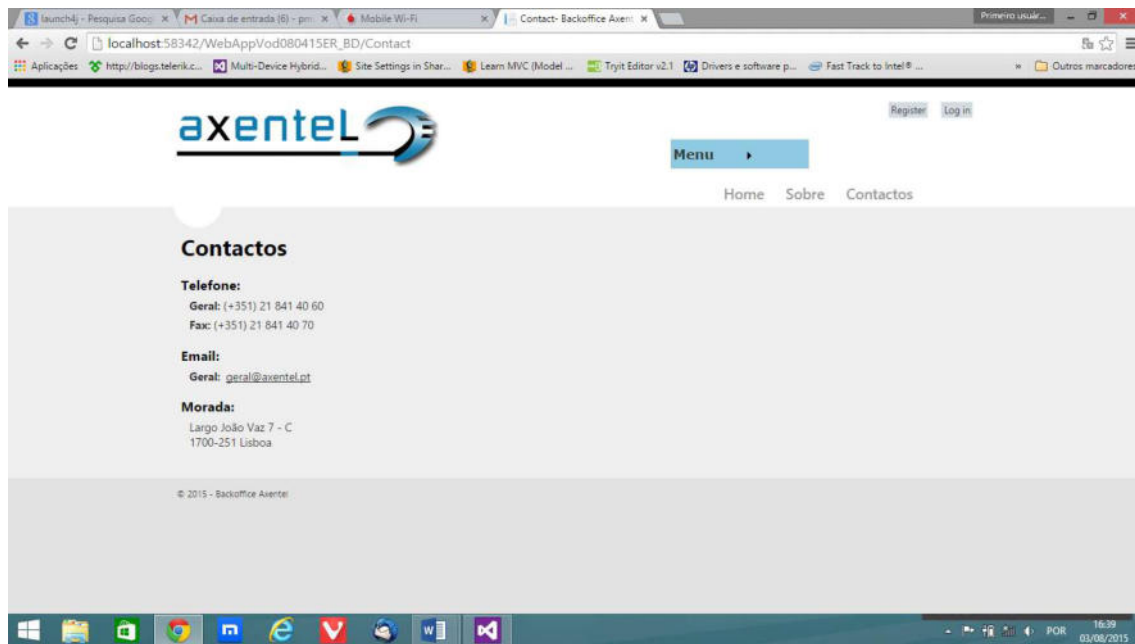


Figura 27 - Página de contactos

7.2.1.3. Página Sobre

A aqui consta informação sobre a empresa (Figura 28).

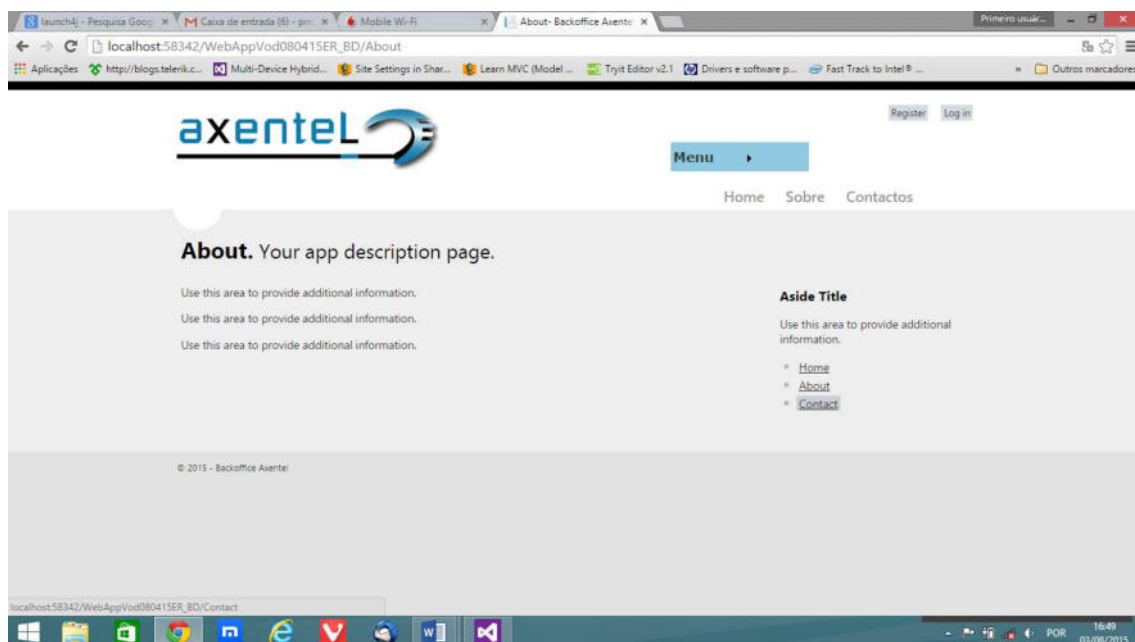


Figura 28 - Página de informação sobre a empresa

7.2.2. Zona reservada

A zona reservada tem 3 níveis de acesso, consoante o tipo de utilizador:

- *Backoffice*: estes utilizadores distribuem o trabalho aos técnicos e submetem ordens de serviço.
- Administração: este grupo possui todas as permissões de acesso à área reservada, ou seja, tem acesso a todas as páginas do *website*. É o grupo que acede a todos os recursos.
- Técnicos: Quando o técnico se autentica tem acesso às suas ordens de serviço (WO) e aos equipamentos que estão na sua posse, pode alocar os equipamentos à respetiva WO quando finaliza a instalação do serviço ADSL na casa do cliente.

A estrutura de navegação da área reservada encontra-se acessível através do botão “menu”.

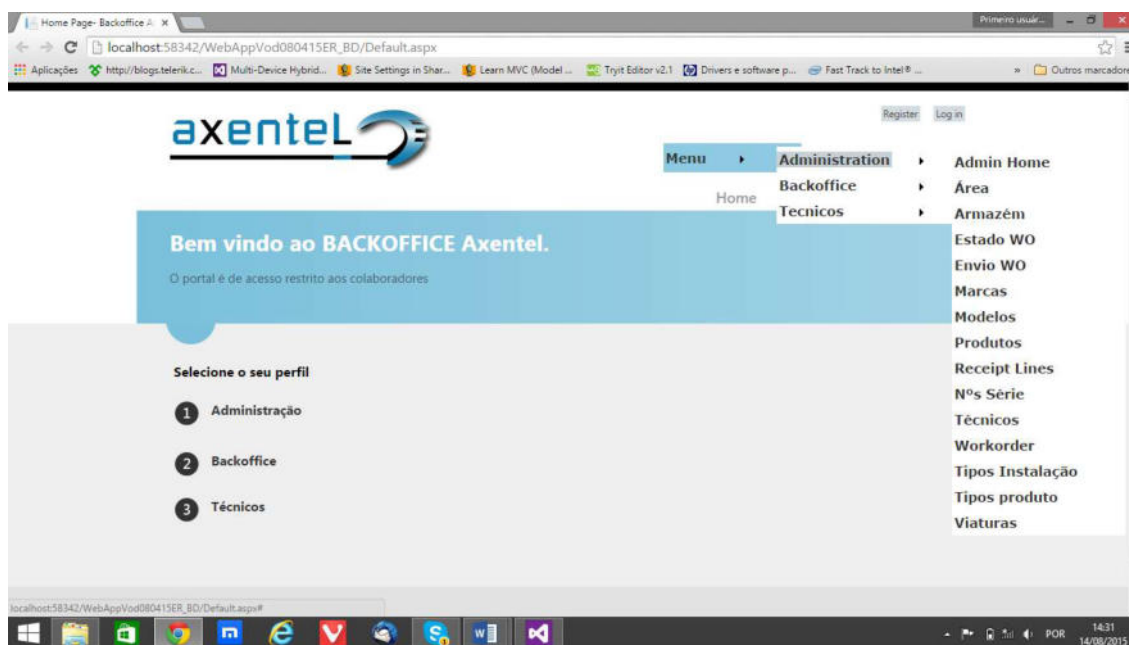


Figura 29 - Acesso área reservada

Existem páginas da área reservada que permitem manusear as diversas tabelas da base de dados como é o caso da área, armazém, marcas, modelos, produtos, n°s de série, técnicos, tipos de instalação, tipos de produtos e viaturas. Estas páginas são usadas para criar, alterar ou apagar registos.

A estrutura das páginas da área reservada é muito semelhante entre si. A título de exemplos, a Figura 30 apresenta a página Área, a Figura 31 a página Técnicos e a Figura 32 a página Armazém.

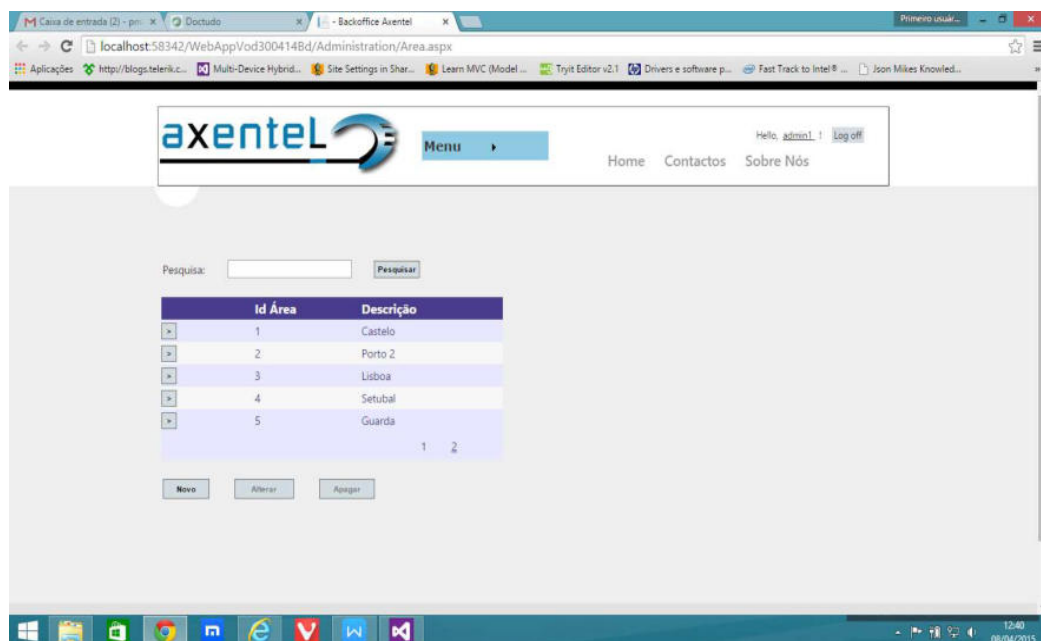


Figura 30 - Página "área"

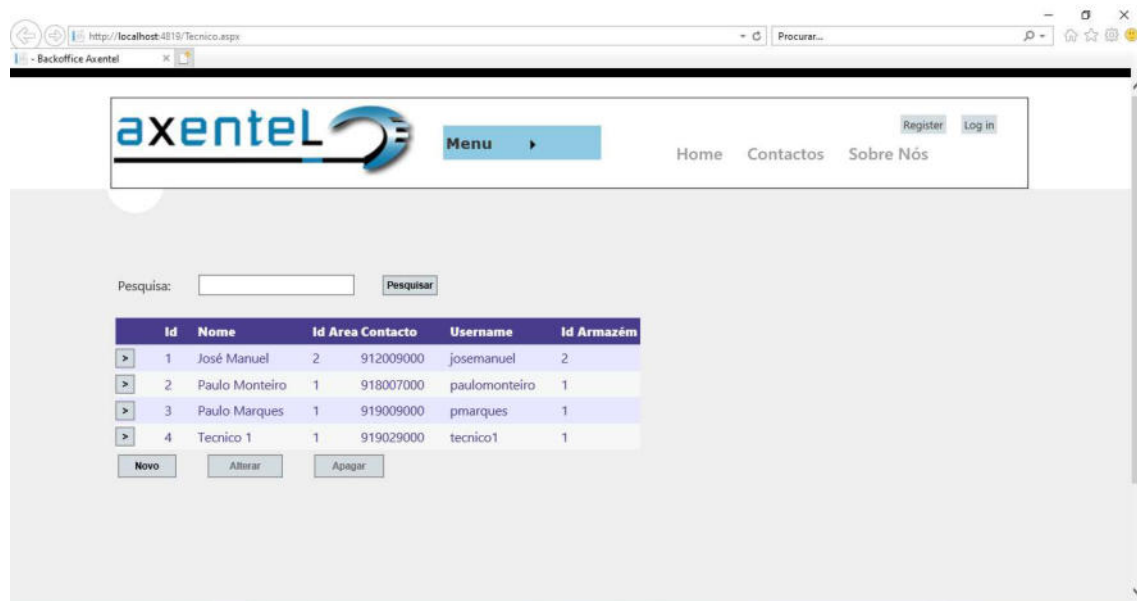


Figura 31 - Página "Técnicos"

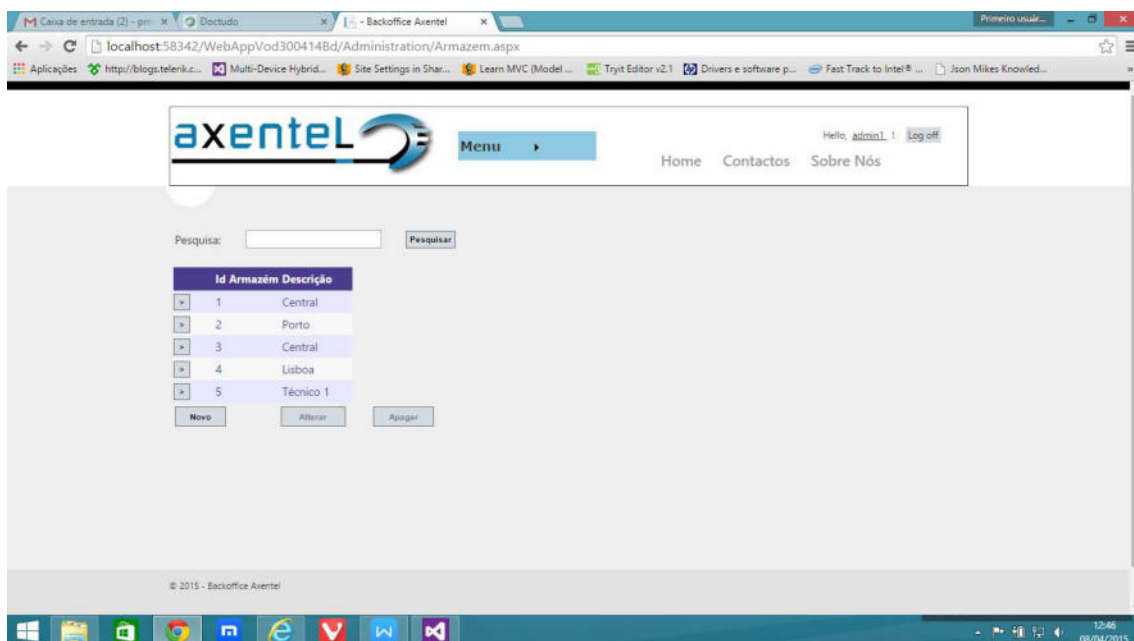


Figura 32 - Página "Armazém"

As páginas “Envio WO” e “*CheckStatus*” são as mais importantes, visto que vão buscar informação a várias tabelas e são as responsáveis por invocar o *web service* da Vodafone. Dada a sua complexidade, estas páginas são descritas de uma forma mais pormenorizada.

1. “Envio WO” - De todas as páginas do *Website* esta é a mais importante e que envolve mais complexidade, pois invoca o método *ConsignStock()*. Este, inicialmente, verifica se o tipo de documento UBL é um *receipt advice* e seguidamente faz uma validação se o elemento *ADSL Id* ou *Access Id* está presente. O documento, se cumprir estes requisitos, passa para fase seguinte de validação de dados, caso contrário é rejeitado de imediato. Aqui a única ação do utilizador é selecionar a WO ou ordem de serviço (*WorkOrder*) que pretende submeter e carregar no botão “Enviar” (Figura 33). Antes de enviar, pode visualizar toda a informação da WO. Após o envio da WO, o utilizador obtém uma mensagem que pode ser de erro ou sucesso, podendo o utilizador corrigir os erros e proceder a novo envio. No caso da mensagem ser de

sucesso, esta não garante que todos os dados enviados sejam válidos, pelo que é sempre necessário verificar o estado da WO posteriormente.

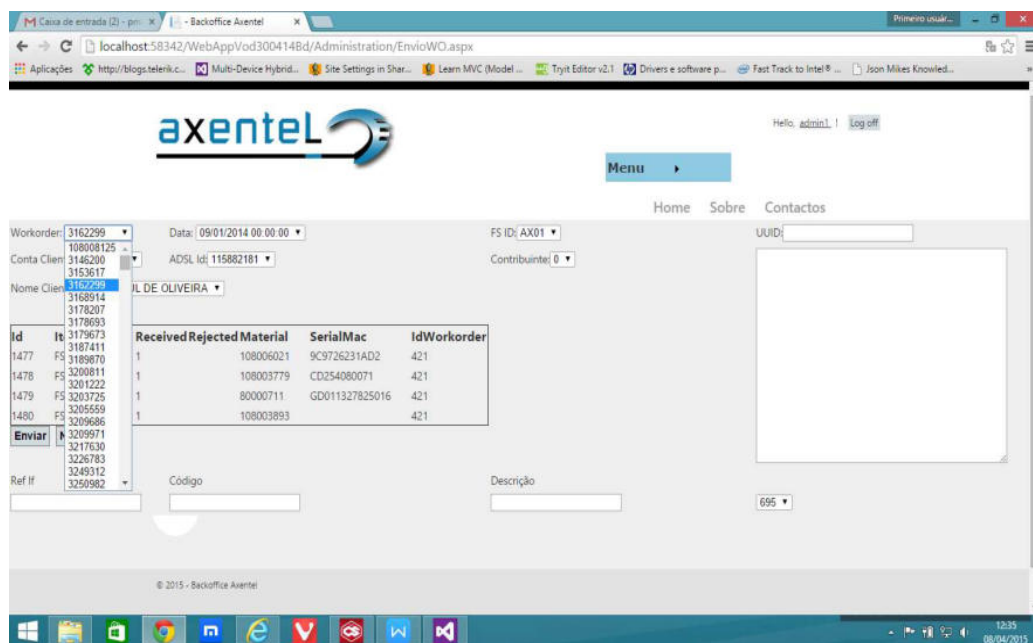


Figura 33 – Página “Envio WO”

2. “*CheckStatus*” - Nesta página é possível saber o estado de uma WO. Existem dois separadores que permitem saber o estado das WO:
 - 2.1. O da esquerda lista todas as WO numa *dropdownlist* após fazer consulta à base de dados, permitindo ao utilizador selecionar a WO pretendida para conhecer o seu estado atual.
 - 2.2. À direita está o separador que foi o utilizado numa fase inicial e que permite que o utilizador introduza manualmente o código da WO que quer validar. Isto dá a liberdade ao utilizador de colocar de forma manual o código da WO e verificar o seu estado. Em ambos os casos, após acionar o botão “Verifica estado”, é invocado o método *checkConsignStatus()* do *Web service* e é obtido o resultado

na caixa de texto situada abaixo do “Estado Atual”. O resultado inclui uma referência, um código e respetiva descrição, que correspondem a um conjunto de erros típicos ou uma simples indicação de sucesso. Na caixa de texto de maiores dimensões e caso o código recebido seja de erro iremos visualizar a descrição do erro de cada linha de produto da WO. A WO é composta por uma ou mais linhas de produtos e estes podem ser *routers*, STB Box, comando e telefones. A Figura 34 mostra a página “*CheckStatus*”.

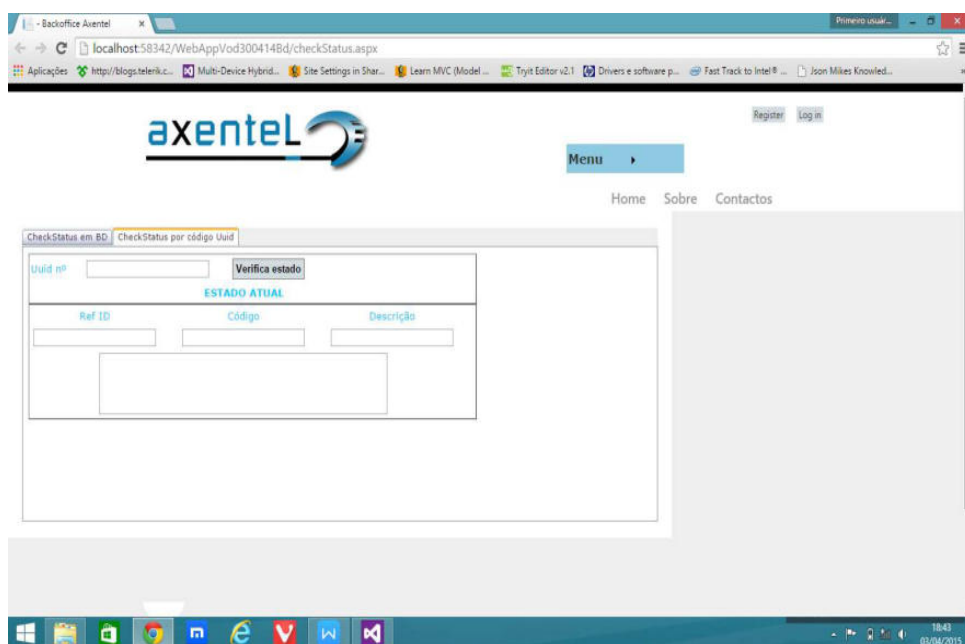


Figura 34 - Página “*Checkstatus*”

Quando foi criada a página “*CheckStatus*”, detetou-se que o documento que nos é devolvido como resposta ao nosso pedido afinal incluía mais informação além da que está descrita na especificação dos métodos do *web service* fornecida pela Vodafone. Então foi criada uma página adicional denominada “*CheckStatusXML*” onde foi incluída uma caixa de texto para visualizar o código XML do documento que é gerado pelo *web service*. Isto permitiu visualizar e tratar informação que nos é remetida pelo *web service*.

“*ReceiptLine*” – Esta página permite ao utilizador adicionar, alterar ou apagar linhas de materiais aplicados nas WO (

3. Figura 35). Pode ainda consultar que materiais constam de uma de terminada WO.

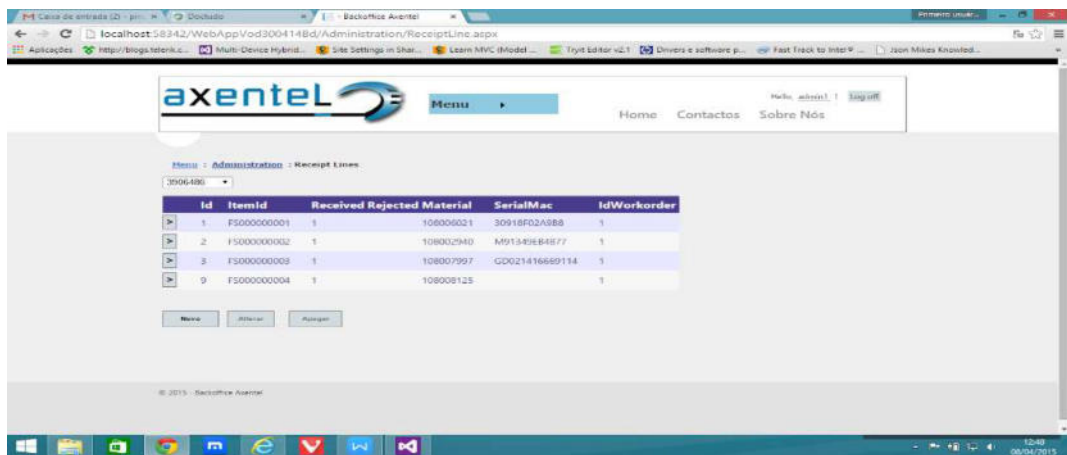


Figura 35 - Página "ReceiptLine"

7.3. Aplicação Web

No *Visual Studio* existem vários tipos de projetos base. Para uma aplicação *web* o mais indicado é o *website* pois disponibiliza uma estrutura estandardizada que permite que o desenvolvimento seja focado no projeto que será trabalhado, deixando de parte código base que é comum em projetos do tipo *website*, como o *webconfig*, e algumas paginas web de raiz que são *default.aspx*, *contact.aspx* e *about.aspx*, que são respetivamente a pagina “inicial”, de “contactos” e “sobre”.

As tarefas deste projeto foram segmentadas da seguinte forma:

1. Integração do *web service* no projeto do *webCPE*.
2. Conversão do código Java desenvolvido. Para aproveitar as soluções encontradas e desenvolvidas no projeto de Java, o código desenvolvido anteriormente vai ser convertido para a linguagem C# que será utilizada nesta aplicação *web*.
3. Desenvolvimento de uma base de dados utilizando o SQL Server.
4. Elaboração do *design* das páginas *web* incluindo os formulários que sejam necessários.

7.4. Integração do *web service* no projeto

A primeira tarefa deste projeto é a integração do ficheiro com a especificação do *web service* da Vodafone. Na implementação do deskCPE esta tarefa foi bastante simples de concretizar, mas tal não sucedeu neste caso.

Após algumas tentativas houve necessidade de aprofundar um pouco mais esta questão, mas como foram encontradas informações divergentes, houve necessidade de testar as vários métodos que são descritos nos próximos pontos.

7.4.1. Método 1 - *Add-reference*

O primeiro método a ser utilizado foi *Add – Service reference*. Este método de integração do *web service* é o que se utiliza normalmente (Microsoft, 2020), coloca-se o endereço do *web service* e este é adicionado ao projeto.

O *web service* da Vodafone exige que quando invocado sejam cumpridos os requisitos de ligação segura através ligação IPsec e enviadas as credenciais de autenticação, sem as quais não é possível a comunicação.

Como neste método não existe forma de colocar as credenciais para que seja obtida a ligação, a mesma não é bem-sucedida e consequentemente não é possível obter a informação necessária com os métodos do *web service*.

O resultado obtido foi nulo conforme representado na Figura 36

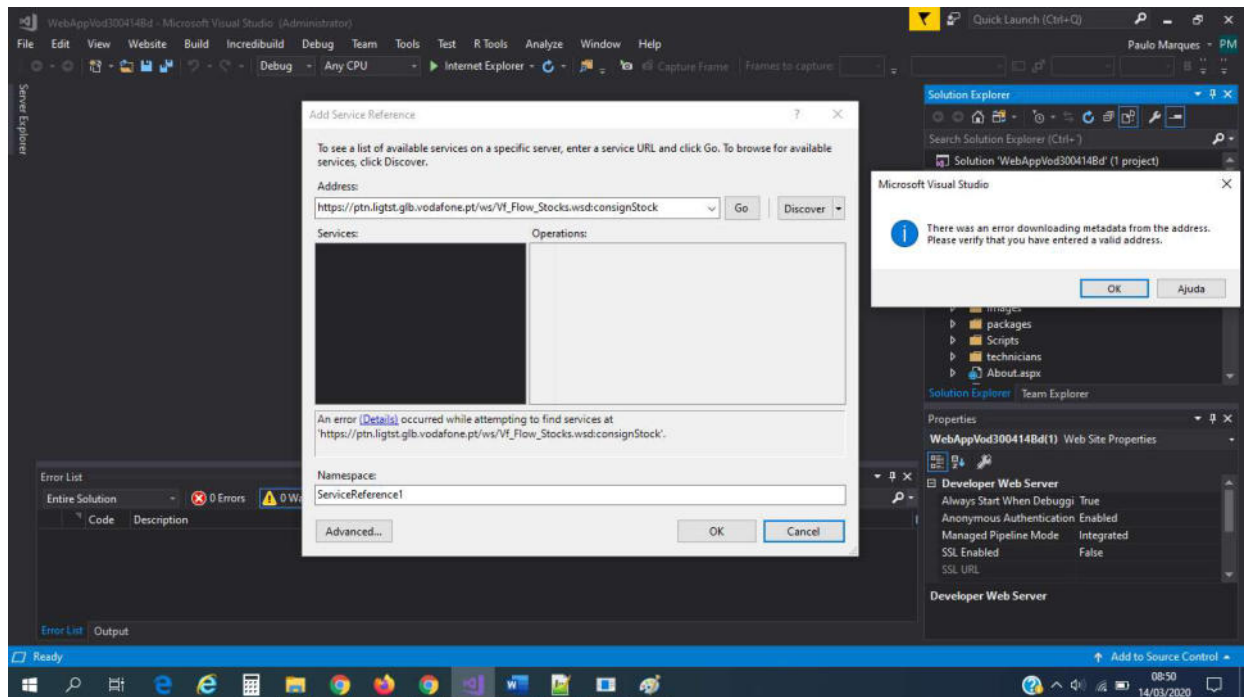


Figura 36 - Add service reference Https

7.4.2. Método 2 - Add-Service reference (Advanced)

Como a Vodafone disponibilizou um ficheiro WSDL com os métodos do web service, existe a possibilidade de utilizar o *Add – Service reference*, mas contrariamente ao efetuado no método 1 não se coloca o endereço do *web service* apenas se seleciona a opção *advanced*. Seguidamente em *Service Reference Settings* premir em *Add Web Reference* e no *URL* colocar a localização física do ficheiro WSDL que é `e:\Vf_Flow2804.WSDL`.

A importação terminava, mas não adicionava nada ao projeto. Verificou-se através pesquisas que o erro estava no ficheiro WSDL e deveria ser alterado. Nos locais onde constava “*parameters*” passar para “*parameter*”. Assim foi possível obter um serviço com os métodos *checkConsignStatus()* e *consignStock()*, e respetivos *datasets* (Figura 37). A situação aparentava estar resolvida. No entanto os objetos UBL não ficaram disponíveis para serem utilizados, pelo que este método também não resultou.

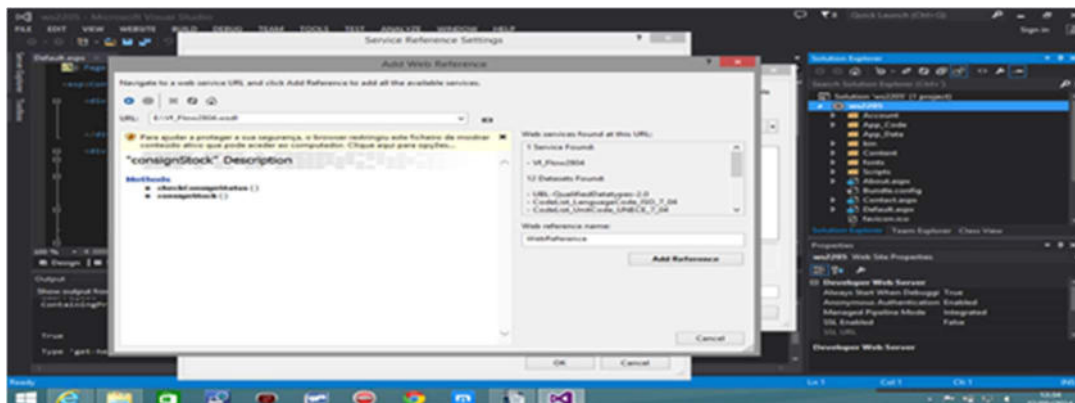


Figura 37 – Add service reference File WSDL

7.4.3. Método 3 - Add-Service reference (Advanced) 2

O procedimento indicado no ponto 2 foi aplicado num novo projeto do tipo *Asp.NET Web Application* e aqui já foi possível replicar as classes anteriormente desenvolvidas em Java para C# porque foram integrados os métodos e classes que permitem a utilização do *web service*, ainda que este processo de conversão das classes não tenha sido linear. Com as classes prontas verifica-se que as classes de interface que permitem invocar o *web service* não foram geradas. Após diversas tentativas e várias pesquisas efetuadas optou-se por abandonar todas as tentativas feitas e procurar uma nova abordagem visto que o problema estava na importação dos métodos do *web service*.

7.4.4. Método 4 - WDSL.exe²

Após análise detalhada dos erros obtidos optou-se por utilizar o comando WDSL.exe que vem incluído no *Visual Studio*. Esta ferramenta é executada em linha de comandos e tem uma grande variedade de opções de acordo com o site da Microsoft (WSDL, 2019). A utilização aparenta ser simples, no entanto neste caso específico tal não aconteceu. Para que o comando seja executado com sucesso é necessário colocar no mesmo diretório do WDSL.exe os ficheiros *XML* essenciais para que os *imports* sejam efetuados corretamente. Ou seja, os elementos necessários para utilização do WDSL.exe devem estar contidos no mesmo diretório, caso contrário pode existir falhas e erros na execução. Os ficheiros *XML* utilizados, todos eles relacionados com a linguagem UBL, foram os seguintes:

² em [http://msdn.microsoft.com/en-us/library/7h3ystb6\(VS.80\).aspx](http://msdn.microsoft.com/en-us/library/7h3ystb6(VS.80).aspx)

- UBL-ApplicationResponse-2.0.xsd
- UBL-CommonAggregateComponents-2.0.xsd
- UBL-CommonBasicComponents-2.0.xsd
- UBL-CommonExtensionComponents-2.0.xsd
- UBL-ExtensionContentDatatype-2.0.xsd
- UBL-QualifiedDatatypes-2.0.xsd
- UBL-ReceiptAdvice-2.0.xsd
- UnqualifiedDataTypeSchemaModule-2.0.xsd
- CodeList_CurrencyCode_ISO_7_04.xsd
- CodeList_MIMEMediaTypeCode_IANA_7_04.xsd
- CodeList_UnitCode_UNECE_7_04.xsd

Finalmente foi executado o comando `C:\>WSDL.exe /language:CSe:\WSDL\Vf_Flow2804.wsdl` e foi obtida a classe *consignStock.cs*, que é o resultado da importação do ficheiro WSDL nesta estão incluídos os métodos que permitem invocar o web service da Vodafone. Desta forma são ultrapassadas todas as dificuldades descritas nos pontos anteriores.

7.5. Sistemas de base de dados e *profiling*

Foi criada uma base de dados dedicada à página *Web*. Como as aplicações são desenvolvidas com ferramentas da Microsoft, optou-se por utilizar o SQL Server 2012 na versão *Express* para sistema de gestão de base de dados. No entanto, esta versão do SQL Server, ainda que adequada a este projeto, apresenta algumas limitações:

- A nível de processador só permite utilizar 1 CPU e *max* de 4 cores.
- Memória RAM só pode utilizar no máximo 1GB.
- O tamanho de bases de dados está limitado a 10GB.

A base de dados da *webCPE* subdivide-se em 2 bases de dados:

- Base de dados CPE, que se encontra representada na Figura 38.
- Base de dados Webservice, representada pela Figura 39.

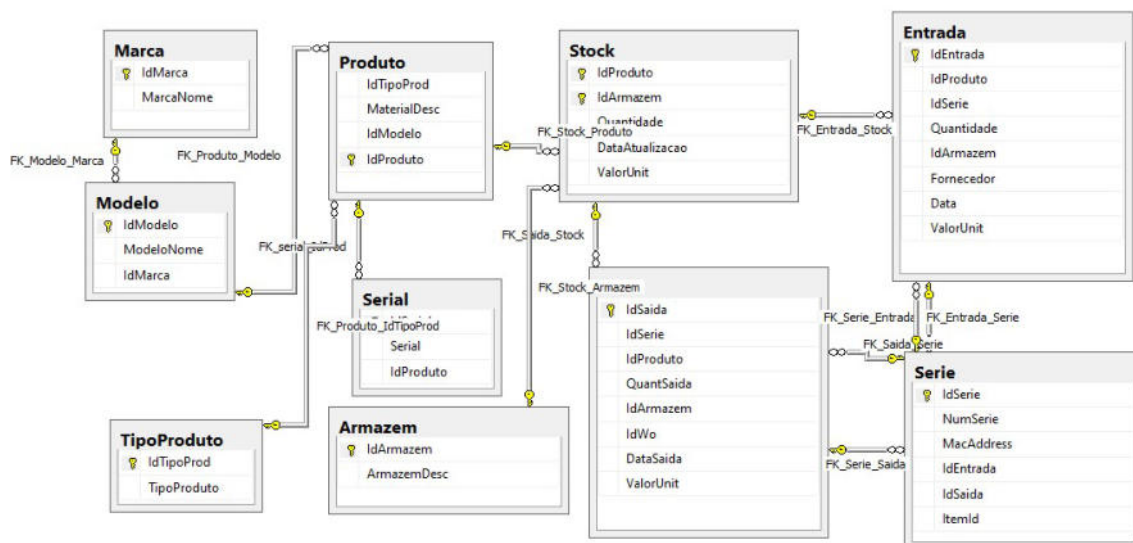


Figura 38 - Tabelas base dados CPE

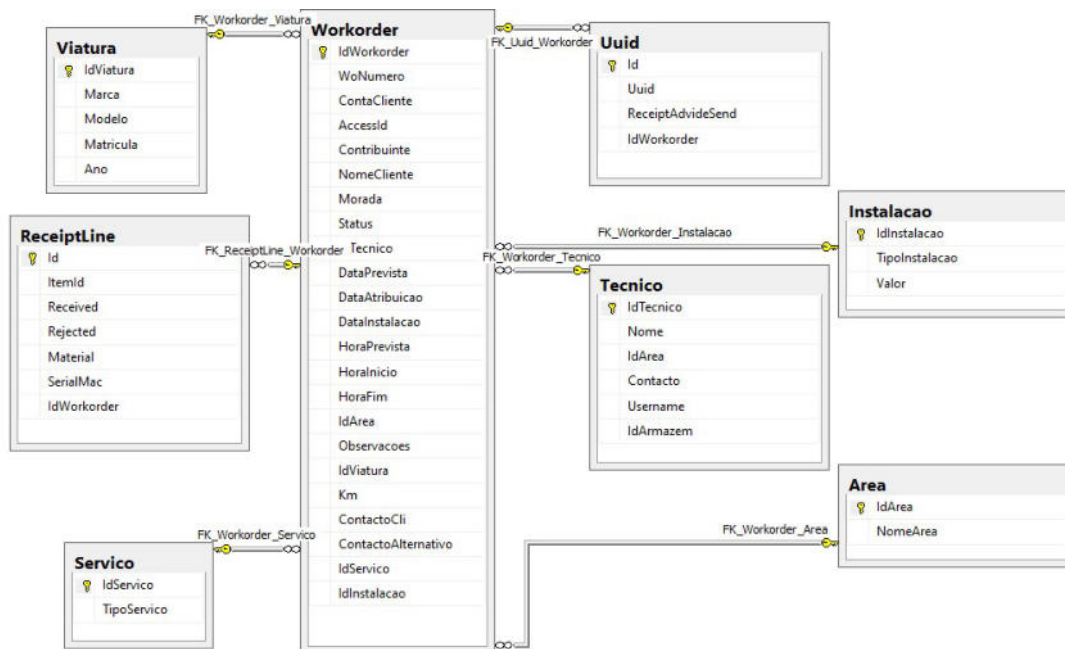


Figura 39 - Tabelas Base dados Webservice

Para otimizar recursos, o *webCPE* irá receber informação de duas bases de dados:

- Da base de dados *webservice* que foi devolvida e que é dedicada à página *Web*;

- Base de dados CPE que suporta o sistema de ERP PHC. Dela é obtida informação de *stocks* de materiais e sua localização, entre outros. Os dados referentes aos equipamentos (*Routers*, TV Box, telefones, entre outros) são introduzidos, quando chegam ao armazém, no PHC. A obtenção de informação através desta base de dados permite que sejam otimizados os recursos humanos pois não será necessário fazer nova introdução de dados na base de dados do website. Em resumo, é utilizada toda a informação que já se encontra no sistema de base de dados do PHC, sendo apenas selecionados os elementos que se necessitam para “alimentar” o *website*. Assim é possível reduzir custos operacionais. O sistema integrado PHC é fechado, não existindo informação sobre o funcionamento e relacionamentos entre tabelas, logo houve necessidade de encontrar forma de obter as consultas pretendidas para a página *Web*. Para tal foi utilizado o *Express Profiler* que é uma ferramenta *open source* semelhante ao *SQL Profiler* da Microsoft.

O *Express Profiler* possui as seguintes características (Sroka, 2016):

- Interface gráfica fácil de utilizar.
- É bastante útil para visualizar como qualquer sistema interage com o SQL, ou seja, vamos considerar um sistema de gestão comercial (SGC) que utiliza o SQL server como sistema de base de dados, o utilizador através do ambiente gráfico do SGC ao acionar um evento, que pode ser gravar, apagar ou editar, no *Express Profiler* será visualizado o comando SQL que o utilizar acionou.
- Permite analisar a complexidade das consultas à base de dados.
- Permite otimização das consultas à base de dados.
- Facilita a verificação de eventuais problemas com as consultas.

O funcionamento do *Express Profiler* é bastante acessível. Coloca-se o endereço do servidor no qual o *SQL Server* está a correr com a autenticação do utilizador SA e é possível visualizar os comandos SQL que o sistema PHC está a enviar. A área assinalada na Figura 40 indica os campos que devem ser preenchidos para efetuar ligação à instância SQL Server.

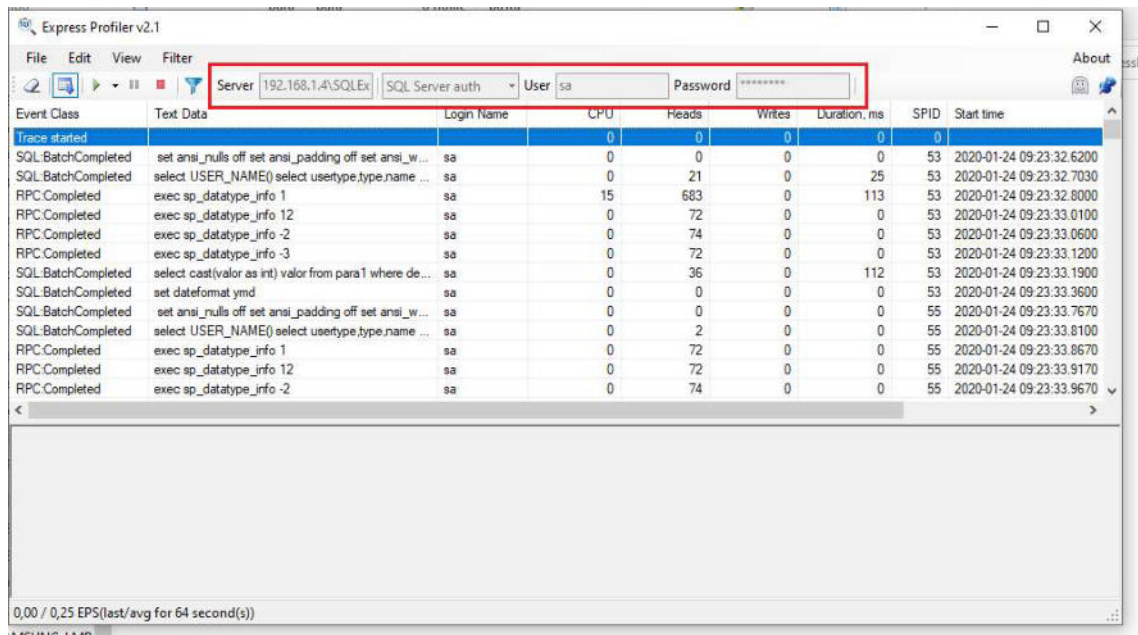
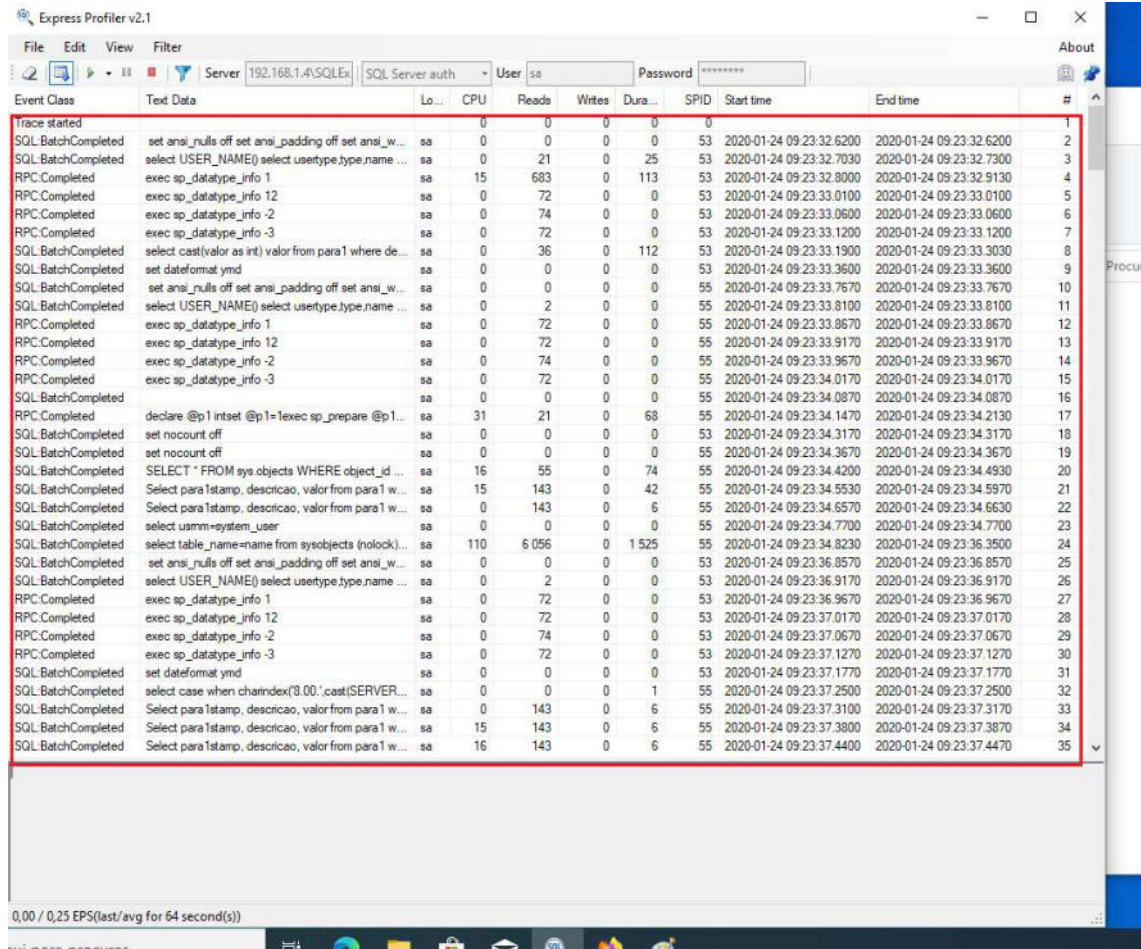


Figura 40 - Express Profiler Login

Após efetuado o *login*, no *express profiler*, à base de dados é possível visualizar todos os “pedidos” que o sistema PHC está a fazer, conforme demonstra a Figura 41.



Express Profiler v2.1

File Edit View Filter

Server: 192.168.1.4\SQLEXP User: sa Password: *****

Event Class	Text Data	Lo...	CPU	Reads	Writes	Dura...	SPID	Start time	End time	#
Trace started			0	0	0	0	0			1
SQL BatchCompleted	set ansi_nulls off set ansi_padding off set ansi_w...	sa	0	0	0	0	53	2020-01-24 09:23:32.6200	2020-01-24 09:23:32.6200	2
SQL BatchCompleted	select USER_NAME() select usertype.type.name ...	sa	0	21	0	25	53	2020-01-24 09:23:32.7030	2020-01-24 09:23:32.7300	3
RPC Completed	exec sp_datatype_info 1	sa	15	683	0	113	53	2020-01-24 09:23:32.8000	2020-01-24 09:23:32.9130	4
RPC Completed	exec sp_datatype_info 12	sa	0	72	0	0	53	2020-01-24 09:23:33.0100	2020-01-24 09:23:33.0100	5
RPC Completed	exec sp_datatype_info -2	sa	0	74	0	0	53	2020-01-24 09:23:33.0600	2020-01-24 09:23:33.0600	6
RPC Completed	exec sp_datatype_info -3	sa	0	72	0	0	53	2020-01-24 09:23:33.1200	2020-01-24 09:23:33.1200	7
SQL BatchCompleted	select cast(valor as int) valor from para1 where de...	sa	0	36	0	112	53	2020-01-24 09:23:33.1900	2020-01-24 09:23:33.3030	8
SQL BatchCompleted	set dateformat ymd	sa	0	0	0	0	53	2020-01-24 09:23:33.3600	2020-01-24 09:23:33.3600	9
SQL BatchCompleted	set ansi_nulls off set ansi_padding off set ansi_w...	sa	0	0	0	0	55	2020-01-24 09:23:33.7670	2020-01-24 09:23:33.7670	10
SQL BatchCompleted	select USER_NAME() select usertype.type.name ...	sa	0	2	0	0	55	2020-01-24 09:23:33.8100	2020-01-24 09:23:33.8100	11
RPC Completed	exec sp_datatype_info 1	sa	0	72	0	0	55	2020-01-24 09:23:33.8670	2020-01-24 09:23:33.8670	12
RPC Completed	exec sp_datatype_info 12	sa	0	72	0	0	55	2020-01-24 09:23:33.9170	2020-01-24 09:23:33.9170	13
RPC Completed	exec sp_datatype_info -2	sa	0	74	0	0	55	2020-01-24 09:23:33.9670	2020-01-24 09:23:33.9670	14
RPC Completed	exec sp_datatype_info -3	sa	0	72	0	0	55	2020-01-24 09:23:34.0170	2020-01-24 09:23:34.0170	15
SQL BatchCompleted		sa	0	0	0	0	55	2020-01-24 09:23:34.0870	2020-01-24 09:23:34.0870	16
RPC Completed	declare @p1 intset @p1=1exec sp_prepare @p1...	sa	31	21	0	68	55	2020-01-24 09:23:34.1470	2020-01-24 09:23:34.2130	17
SQL BatchCompleted	set nocount off	sa	0	0	0	0	53	2020-01-24 09:23:34.3170	2020-01-24 09:23:34.3170	18
SQL BatchCompleted	set nocount off	sa	0	0	0	0	55	2020-01-24 09:23:34.3670	2020-01-24 09:23:34.3670	19
SQL BatchCompleted	SELECT * FROM sys.objects WHERE object_id ...	sa	16	55	0	74	55	2020-01-24 09:23:34.4200	2020-01-24 09:23:34.4930	20
SQL BatchCompleted	Select para1stamp, descricao, valor from para1 w...	sa	15	143	0	42	55	2020-01-24 09:23:34.5530	2020-01-24 09:23:34.5970	21
SQL BatchCompleted	Select para1stamp, descricao, valor from para1 w...	sa	0	143	0	6	55	2020-01-24 09:23:34.6570	2020-01-24 09:23:34.6630	22
SQL BatchCompleted	select uamm-system_user	sa	0	0	0	0	55	2020-01-24 09:23:34.7700	2020-01-24 09:23:34.7700	23
SQL BatchCompleted	select table_name=name from sysobjects (nolock)...	sa	110	6 056	0	1 525	55	2020-01-24 09:23:34.8230	2020-01-24 09:23:36.3500	24
SQL BatchCompleted	set ansi_nulls off set ansi_padding off set ansi_w...	sa	0	0	0	0	53	2020-01-24 09:23:36.8570	2020-01-24 09:23:36.8570	25
SQL BatchCompleted	select USER_NAME() select usertype.type.name ...	sa	0	2	0	0	53	2020-01-24 09:23:36.9170	2020-01-24 09:23:36.9170	26
RPC Completed	exec sp_datatype_info 1	sa	0	72	0	0	53	2020-01-24 09:23:36.9670	2020-01-24 09:23:36.9670	27
RPC Completed	exec sp_datatype_info 12	sa	0	72	0	0	53	2020-01-24 09:23:37.0170	2020-01-24 09:23:37.0170	28
RPC Completed	exec sp_datatype_info -2	sa	0	74	0	0	53	2020-01-24 09:23:37.0670	2020-01-24 09:23:37.0670	29
RPC Completed	exec sp_datatype_info -3	sa	0	72	0	0	53	2020-01-24 09:23:37.1270	2020-01-24 09:23:37.1270	30
SQL BatchCompleted	set dateformat ymd	sa	0	0	0	0	53	2020-01-24 09:23:37.1770	2020-01-24 09:23:37.1770	31
SQL BatchCompleted	select case when charindex(8.00'.cast(SERVER...	sa	0	0	0	1	55	2020-01-24 09:23:37.2500	2020-01-24 09:23:37.2500	32
SQL BatchCompleted	Select para1stamp, descricao, valor from para1 w...	sa	0	143	0	6	55	2020-01-24 09:23:37.3100	2020-01-24 09:23:37.3170	33
SQL BatchCompleted	Select para1stamp, descricao, valor from para1 w...	sa	15	143	0	6	55	2020-01-24 09:23:37.3800	2020-01-24 09:23:37.3870	34
SQL BatchCompleted	Select para1stamp, descricao, valor from para1 w...	sa	16	143	0	6	55	2020-01-24 09:23:37.4400	2020-01-24 09:23:37.4470	35

0,00 / 0,25 EPS(last/avg for 64 second(s))

Figura 41 – Express profiler output

Assim, quando se interage com o PHC de uma forma visual é possível aceder ao comando SQL que é enviado do PHC para o *SQL Server* (Figura 42). Por exemplo, pretende-se verificar quais os equipamentos de determinada referência que estão no armazém, é possível através de um sistema de consultas existente no PHC conseguirmos obter uma listagem desses equipamentos, isto tudo de uma forma bastante acessível ao utilizador normal do PHC.

Event Class	Text Data	Login Name	CPU	Reads	Writes	Duration, ms	SPID	Start time
RPC:Completed	exec sp_executesql N'SELECT * FROM DBO.EX...	sa	0	0	0	0	55	2020-01-24 09:24:59.0130
RPC:Completed	exec sp_executesql N'SELECT * FROM DBO.EX...	sa	0	0	0	0	55	2020-01-24 09:24:59.0730
RPC:Completed	exec sp_executesql N'SELECT * FROM DBO.EX...	sa	0	0	0	0	55	2020-01-24 09:24:59.1300
SQL:BatchCompleted	Select top 1 MAQUINA as campo from MA (noloc...	sa	0	21	0	57	55	2020-01-24 09:25:02.6470
SQL:BatchCompleted	SELECT tc.ecran, ti.* FROM ti (NOLOCK) INNER...	sa	0	40	0	6	55	2020-01-24 09:25:02.8470
SQL:BatchCompleted	Select distinct MAQUINA as campo from MA (nol...	sa	203	3 226	0	867	55	2020-01-24 09:25:08.5200
SQL:BatchCompleted	SELECT tc.ecran, ti.* FROM ti (NOLOCK) INNER...	sa	0	40	0	6	55	2020-01-24 09:25:09.6970
SQL:BatchCompleted	SELECT tc.ecran, ti.* FROM ti (NOLOCK) INNER...	sa	0	21	0	0	55	2020-01-24 09:25:29.3800
SQL:BatchCompleted	SELECT tc.ecran, ti.* FROM ti (NOLOCK) INNER...	sa	0	21	0	0	55	2020-01-24 09:25:31.6930
SQL:BatchCompleted	SELECT tc.ecran, ti.* FROM ti (NOLOCK) INNER...	sa	0	21	0	0	55	2020-01-24 09:25:34.8800
SQL:BatchCompleted	select MA.MAstamp,MA.SERIE,MA.SERIE2,MA...	sa	78	3 722	0	329	55	2020-01-24 09:25:36.3930
SQL:BatchCompleted	SELECT tc.ecran, ti.* FROM ti (NOLOCK) INNER...	sa	0	21	0	0	55	2020-01-24 09:25:37.0400
SQL:BatchCompleted	SELECT tc.ecran, ti.* FROM ti (NOLOCK) INNER...	sa	0	21	0	0	55	2020-01-24 09:25:41.1530
Trace paused			0	0	0	0	0	

```

select MA.MAstamp,MA.SERIE,MA.SERIE2,MA.MARCA,MA.MAQUINA,MA.TIPO,MA.NOME,MA.AGNOME,MA.ARMATAIN from MA (nolock) where MA.MAQUINA
LIKE '%DP750%' and MA.NOARM = 1 and MA.CASA = 1
go

```

0,00 / 0,25 EPS(last/avg for 64 second(s))

Figura 42 - Express profiler query

Do lado do *website* são necessários os dados que resultam desta listagem, mas sob a forma de uma *query* em SQL para depois serem adicionados à base de dados do *website*.

O *express profiler* permite ainda filtrar a informação obtida, conforme apresentado na Figura 43.

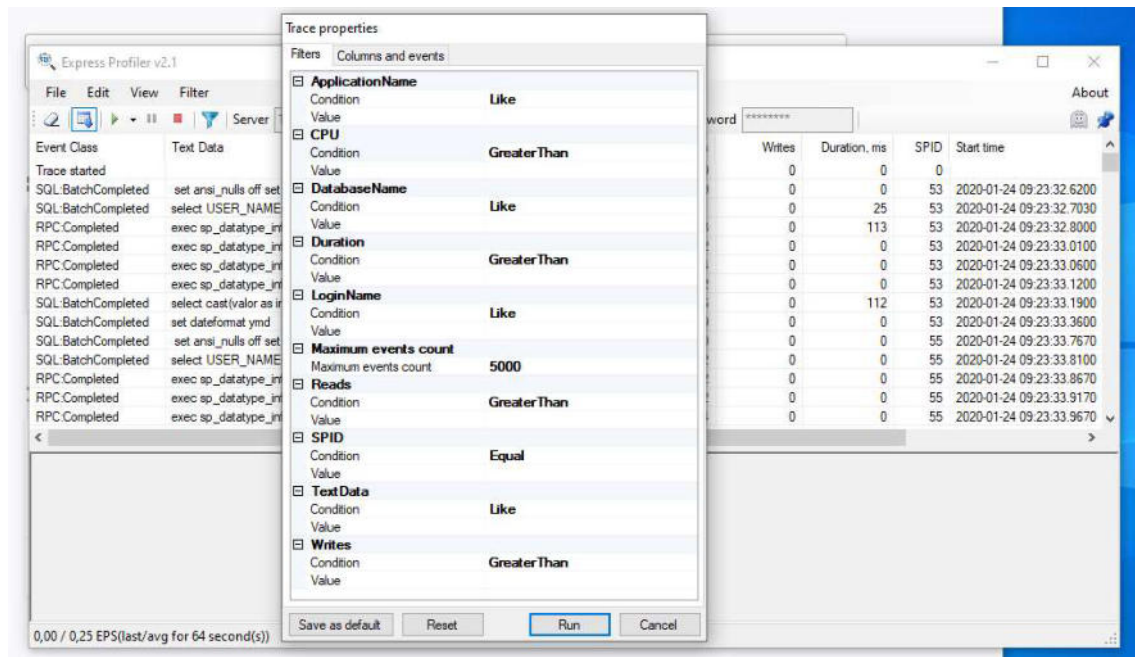


Figura 43 - Filtros Express Profiler

O que o *express profiler* disponibiliza é importante para perceber como interagir com a base de dados, sem que seja necessário saber como as tabelas desta se relacionam entre si. A lógica do funcionamento da base de dados é disponibilizada pelo *output* do *express profiler*.

A utilização e alteração das linhas de comandos SQL obtidas deve ser alvo de uma análise cuidadosa antes de serem utilizadas para interagir com a base de dados. No caso dos comandos de *Select*, como não fazem alterações nas bases de dados, podem ser utilizados de uma forma mais leviana, no entanto, quando se pretende fazer um *insert*, *update* ou *delete* devem-se tomar precauções, pois com uma ação destas, a base de dados poderá perder a sua integridade.

Foram criados dois novos *webservices* para interagir com as bases de dados existentes: um para efetuar as consultas à base de dados do PHC e outro que servirá de suporte às consultas que a aplicação móvel faz à base de dados do *website*.

7.6. Segurança do sistema

O acesso à área reservada do *website* foi configurado com a ferramenta “*ASP.net configuration*”, incluída no *VisualStudio*, já que esta permite criar *roles* e perfis de utilizador e associá-los aos utilizadores. Esta funcionalidade está disponível através do menu Project conforme indicado na Figura 44.

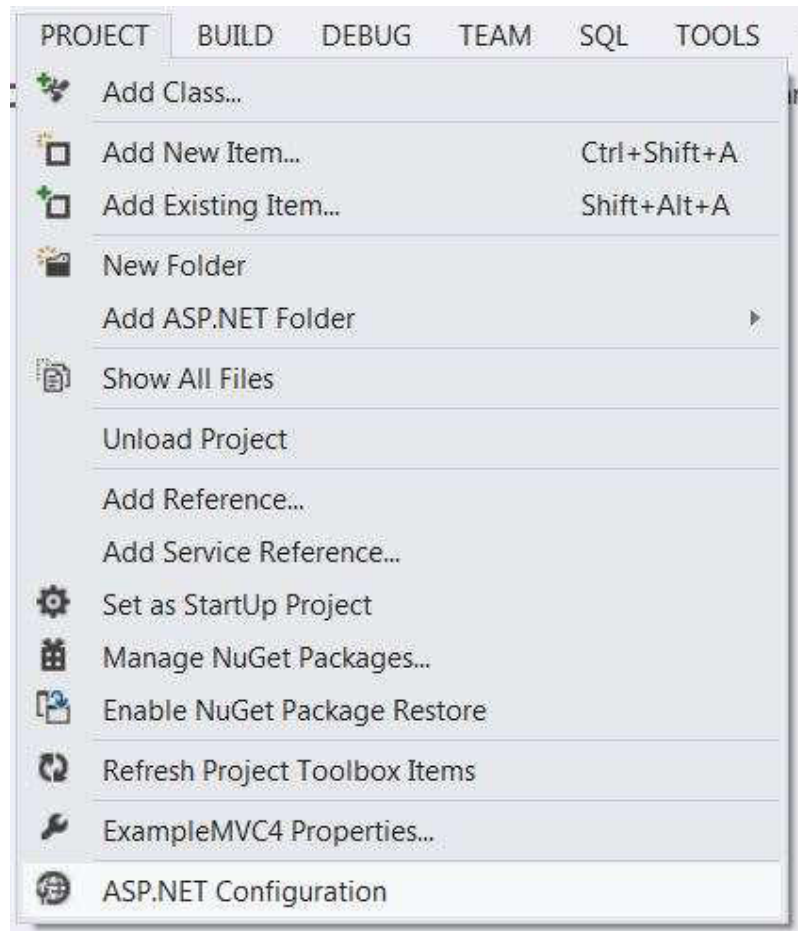
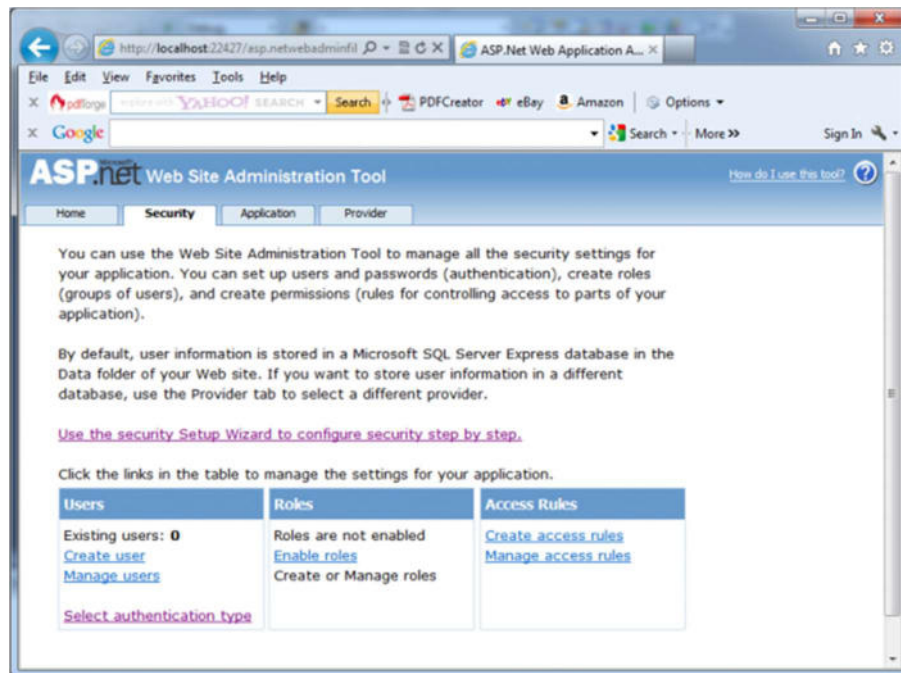


Figura 44 - Asp.Net Configuration

É assim obtido acesso à página na qual é possível gerir os *roles* e *users* (Figura 45). Conforme já indicado no ponto 7.2.2, foram criados três níveis de acesso ao *Website* que correspondem aos *roles* criados aqui. Os *users* são associados aos *roles* e assim se obtém uma gestão de acesso.

Figura 45 - Gestão de *roles* e *users*

Um elemento que deve ser encriptado ou omitido são as *connection strings* (descrição das ligações às bases de dados). Esta informação contém dados de carácter confidencial, pois inclui o endereço IP do servidor, respetiva porta, utilizador e palavra passe com direitos de proprietário. Este conjunto de informação não pode estar exposta no código pois irá resultar numa falta de segurança que pode por em causa a integridade dos dados.

Este problema pode ser ultrapassado utilizando uma ferramenta de *Data Protection API* (DPAPI) ou RSA, para encriptar informação sensível que consta no *web.config*.

O DPAPI vem incluído no Windows e utiliza *Keys* específicas para determinada máquina ou servidor. Esta é uma boa solução quando apenas é utilizado um único servidor, quando se pretende utilizar esta *Key* noutros servidores a solução não funciona.

Para as *Keys* serem utilizadas em várias máquinas ou servidores tem de ser utilizada a RSA. Como pretendemos que a encriptação seja utilizada em diferentes servidores, foi utilizado o *RsaProtectedConfigurationProvider* que é um *provider Data protection* que é invocado pelo *aspnet_regiis* que faz parte da *framework* do Microsoft.NET. Neste caso foi utilizado de acordo com os passos seguintes:

1 - Em linha de comandos, ir até ao diretório `c:\ Windows\ Microsoft.NET\ Framework\ v4.0.30319`

2 - Criar o contentor da chave RSA, `aspnet_regiis -pc "minhaRsa" -exp`

3 - No `web.config` adicionar o seguinte código:

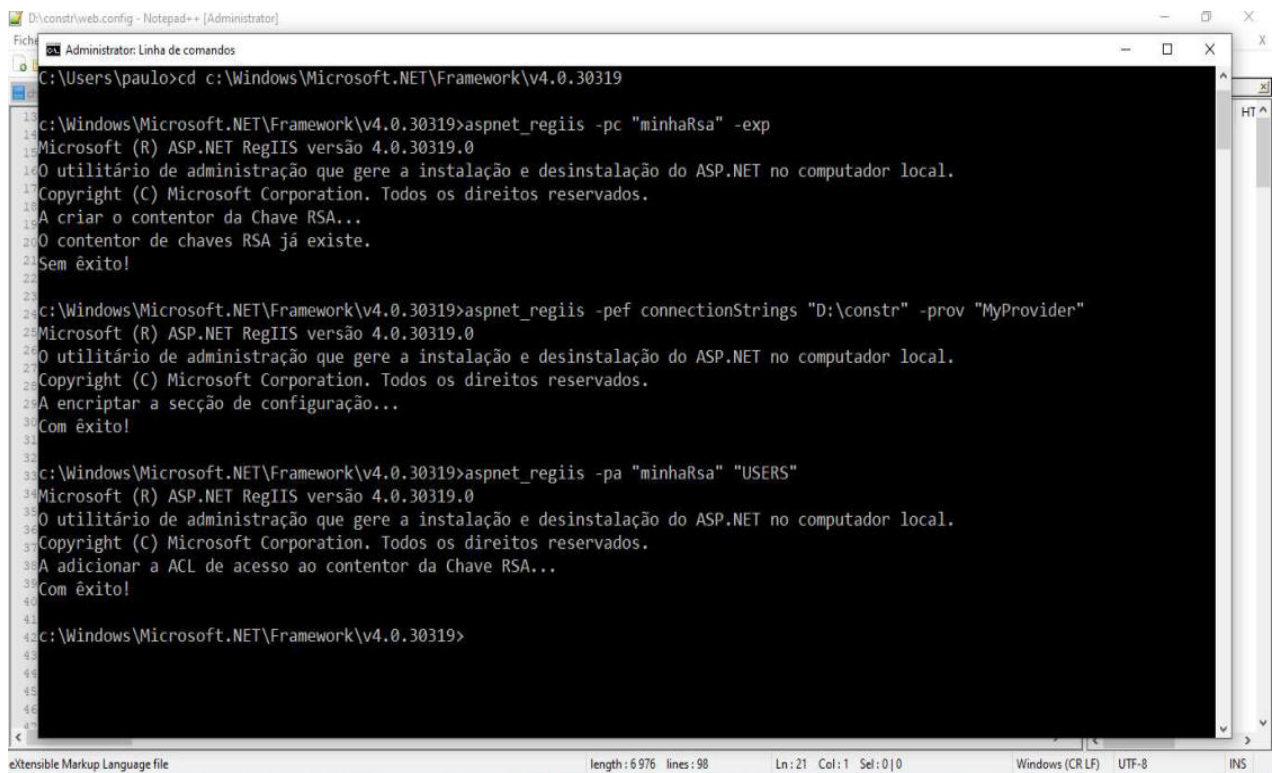
```
<configProtectedData>
  <providers>
    <add name="MyProvider"
      type="System.Configuration.RsaProtectedConfigurationProvider, System.Configuration,
        Version=2.0.0.0,&#xD;&#xA;
        Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a,&#xD;&#xA;
        processorArchitecture=MSIL"
      keyContainerName="minhaRsa"
      useMachineContainer="true" />
  </providers>
</configProtectedData>
```

4 - executar `aspnet_regiis -pef connectionStrings "D:\constr" -prov "MyProvider"`

5 - Permitir acesso ao contentor da chave RSA, `aspnet_regiis -pa "minhaRsa" "USERS"`. Exemplificado na Figura 46.

Existe a possibilidade de descriptar o o ficheiro para a sua forma original, sem encriptação, esta funcionalidade é útil para quando se pretende efetuar alterações no ficheiro. Para descriptar é executado o seguinte comando:

1 - `aspnet_regiis -pdf connectionStrings "D:\constr"`.



```
D:\constr\web.config - Notepad++ [Administrator]
Administrator: Linha de comandos
C:\Users\paulo>cd c:\Windows\Microsoft.NET\Framework\v4.0.30319
c:\Windows\Microsoft.NET\Framework\v4.0.30319>aspnet_regiis -pc "minhaRsa" -exp
Microsoft (R) ASP.NET RegIIS versão 4.0.30319.0
O utilitário de administração que gere a instalação e desinstalação do ASP.NET no computador local.
Copyright (C) Microsoft Corporation. Todos os direitos reservados.
A criar o contentor da Chave RSA...
O contentor de chaves RSA já existe.
Sem êxito!
c:\Windows\Microsoft.NET\Framework\v4.0.30319>aspnet_regiis -pef connectionStrings "D:\constr" -prov "MyProvider"
Microsoft (R) ASP.NET RegIIS versão 4.0.30319.0
O utilitário de administração que gere a instalação e desinstalação do ASP.NET no computador local.
Copyright (C) Microsoft Corporation. Todos os direitos reservados.
A encriptar a secção de configuração...
Com êxito!
c:\Windows\Microsoft.NET\Framework\v4.0.30319>aspnet_regiis -pa "minhaRsa" "USERS"
Microsoft (R) ASP.NET RegIIS versão 4.0.30319.0
O utilitário de administração que gere a instalação e desinstalação do ASP.NET no computador local.
Copyright (C) Microsoft Corporation. Todos os direitos reservados.
A adicionar a ACL de acesso ao contentor da Chave RSA...
Com êxito!
c:\Windows\Microsoft.NET\Framework\v4.0.30319>
```

Figura 46 - Encriptação

A encriptação do ficheiro *app config* permite salvaguardar informação sensível. A solução apresentada foi uma das mais interessantes encontrada após várias buscas efetuadas na Internet. Esta pode ser consultada no endereço <http://dotnetcodr.com/2013/11/18/how-to-protect-your-config-file-in-net-with-cryptography/> (Nemes, 2013). As outras soluções passavam por “esconder” as *connections strings* em ficheiros encriptados, mas que poderiam ser desencriptados facilmente ou então consistia em desenvolver um *web service* para guardar as *connection strings*.

8. APLICAÇÃO PARA DISPOSITIVOS MÓVEIS (*SMARTPHONES*)

De acordo com os dados disponíveis na Tabela 1, de 2014 até 2018 prevê-se um crescimento significativo do número de utilizadores de dispositivos móveis, adicionalmente outro indicador é a percentagem de população mundial que utiliza um smartphone, de 2016 a 2020 espera-se que atinja um crescimento superior a 11 %, conforme indicado na Tabela 2 (Statista, 2020) Tabela 2 - Global smartphone penetration. Outro indicador importante para o mercado dos dispositivos e desenvolvimento de aplicações móveis é o impacto que terão nas vendas online ou e-commerce conforme se verifica na Tabela 3. Tendo em atenção os fatores de crescimento indicados a oferta, por parte dos fabricantes, de dispositivos móveis quer sejam telemóveis denominados *smartphones* ou *tablets* irá também aumentar. No desenvolvimento de plataformas ou sistemas operativos houve uma evolução positiva nomeadamente com surgimento da plataforma *Android* que se veio juntar às já existentes *Windows Phone* da *Microsoft* e ao *iOS* da *Apple*. Além destas plataformas existem outras que não conseguiram atingir o mercado da mesma forma como é o caso do *BlackBerry* (Chang, 2020).

Worldwide Smartphone Forecast by Region, Shipments, Market Share and 5-Year CAGR (units in millions)

Operating System	2014 Shipment Volumes*	2014 Market Share	2018 Shipment Volumes*	2018 Market Share	2013-2018 CAGR
Android	997.7	80.2%	1,401.3	77.6%	12.0%
iOS	184.1	14.8%	247.4	13.7%	10.0%
Windows Phone	43.3	3.5%	115.3	6.4%	28.1%
BlackBerry	9.7	0.8%	4.6	0.3%	-25.0%
Others	9.3	0.7%	37.7	2.1%	31.5%
Total	1,244.1	100%	1,806.3	100%	12.3%

Source: IDC Worldwide Mobile Phone Tracker, May 28, 2014

* Forecast data

Tabela 1 - Mobile Forecast 2014

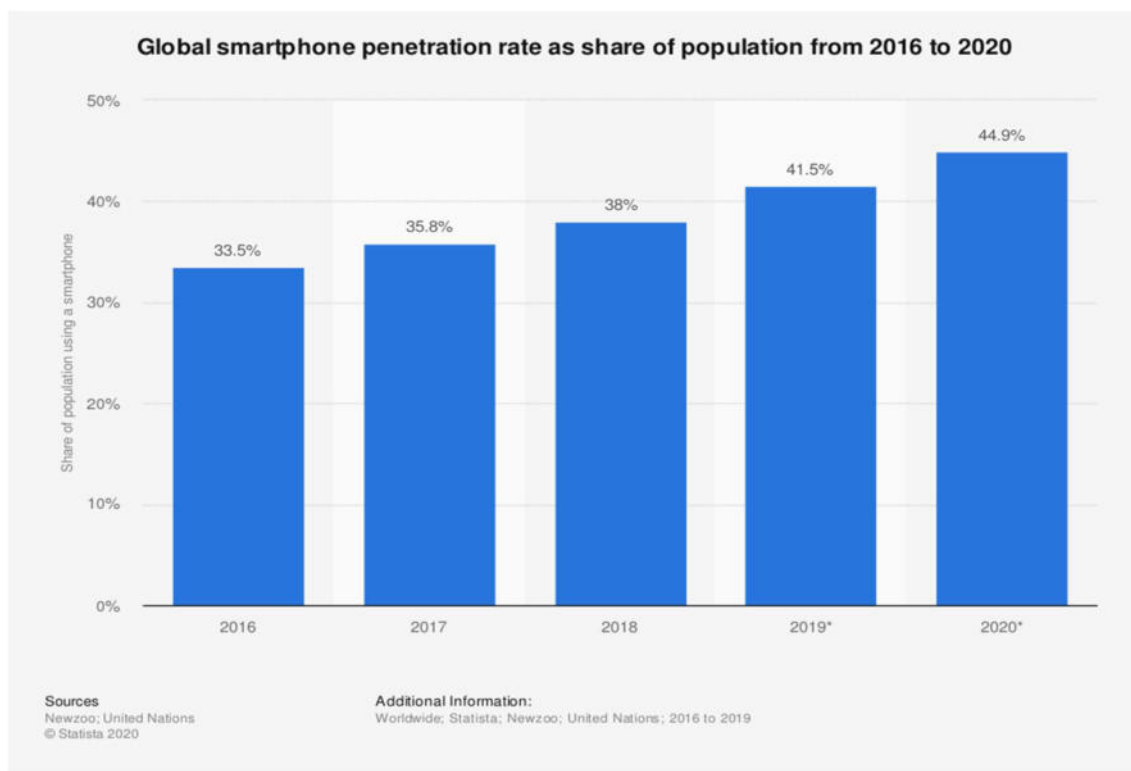


Tabela 2 - Global smartphone penetration

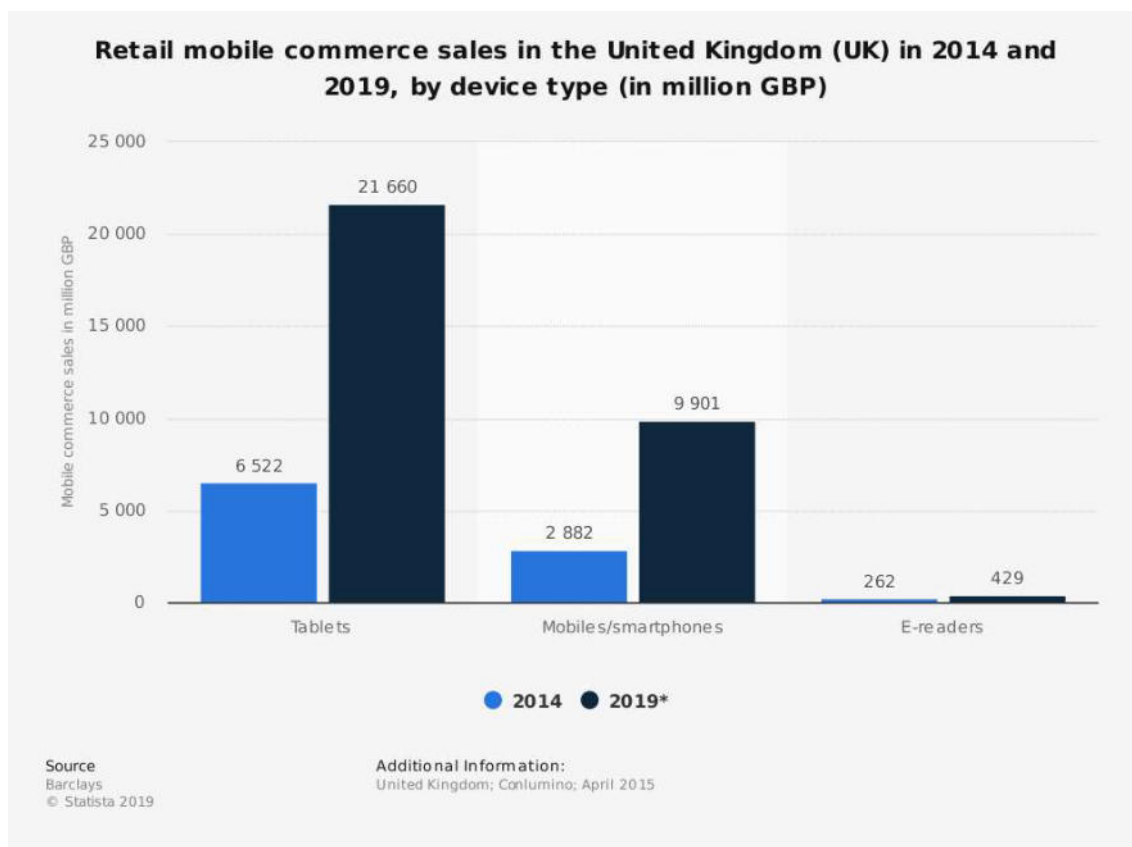


Tabela 3 - Retail mobile commerce sales

Perante este cenário, a plataforma *Android* é e será a mais utilizada, no entanto tem de ser feita uma avaliação mais objetiva considerando a realidade da Atena T. Surgem então as seguintes questões:

- Quais as características dos *smartphones* que a empresa disponibilizou aos utilizadores?

- Que plataforma utilizam? *iOS*, *Android* ou *Windows phone*?

Como seria de esperar, e após reunião com a administração da Atena T, foi demonstrado o interesse em desenvolver uma aplicação multiplataforma, pois existem equipamentos bastante divergentes quer em características quer em tipo de plataforma. Verificou-se ainda que os técnicos utilizam simultaneamente portáteis com ambiente *Windows* e *smartphones* com sistema *Android*. Mas existem ainda alguns utilizadores internos com equipamentos *IOS* e *Windows Phone* e um *tablet* com sistema operativo *Windows*.

Uma solução passaria por desenvolver uma aplicação para cada plataforma (*Android* utilizando o *Netbeans* ou *Eclipse*, *iOS* com o *Xcode*, *Windows Phone/Windows* com o *VisualStudio*). A ideia de desenvolver uma aplicação multiplataforma surge como sendo a mais interessante. No entanto a utilização de uma ferramenta multiplataforma possui algumas desvantagens (Luis Corral, 2012) como:

- O desenvolvimento aplicativo, não sendo nativo, não são exploradas todas as potencialidades de sistema específico.
- Poderá não permitir ou limitar acesso a algumas funcionalidades do dispositivo móvel.

Como vantagens podem-se destacar as seguintes (Luis Corral, 2012):

- Um único desenvolvimento pode ser aplicado em plataformas distintas.
- A mesma aplicação pode ser distribuída através dos vários *marketplaces*.
- São agregadas várias linguagens de programação e *frameworks* numa única plataforma de desenvolvimento.

Dentro das soluções de desenvolvimento multiplataforma, foram selecionadas cinco ferramentas de desenvolvimento, o *Titanium Studio*, *Multi-device Hybrid Apps for Visual Studio*, *PhoneGap*, *Intel XDK* e *Responsive Bootstrap*.

8.1. Titanium Studio

Esta ferramenta apresentou-se como sendo inovadora, permitindo o desenvolvimento multiplataforma, criação de aplicações de forma flexível. O que normalmente se faz em vários dias pode ser atingido em apenas algumas horas e poucas linhas de código (Angelini, 2012). Efetuada a instalação do produto e após várias tentativas de configuração, concluiu-se que se tratava de uma ferramenta com vários conflitos e com falta de robustez. Nos testes efetuados, apenas foi possível deixar a ferramenta preparada para desenvolver aplicações para uma plataforma.

Chegou então o momento de abandonar esta solução. Posteriormente verificou-se que a empresa *Appcelerator Inc*, que desenvolveu o *Titanium*, reformulou a ferramenta e passou a cobrar o seu uso.

8.2. Multi-device Hybrid Apps for Visual Studio

Em meados de 2014 surge o *Multi-device Hybrid Apps for Visual Studio* (Microsoft, 2020). Esta ferramenta é apresentada como um complemento do *Visual Studio*, e obviamente apresenta-se como a ferramenta ideal para complementar este projeto, tendo em atenção que grande parte foi desenvolvido com as ferramentas Microsoft. O material disponibilizado pela Microsoft é bastante promissor e inclui tutoriais, vídeos formativos e vasta bibliografia. Após visualização de alguns tutoriais concluiu-se que esta ferramenta ainda se encontrava numa fase muito embrionária.

8.3. PhoneGap

O PhoneGap pertence também ao grupo de ferramentas multiplataforma. As aplicações são desenvolvidas com recurso a HTML5, CSS3 e JavaScript não dependendo de API's específicas. Nesta plataforma, o acesso aos recursos do *smartphone*, como as câmaras, GPS e colunas de som, é efetuado através da tecnologia *Apache Cordova* (Cordova, 2019). As aplicações desenvolvidas são designadas como híbridas pois não são exclusivamente *web* nem nativas, ou seja, têm acesso às API nativas dos sistemas Android, IOS e Windows Phone, isto porque o PhoneGap permite criar uma distribuição da aplicação desenvolvida para qualquer um dos sistemas moveis referidos. Em resumo o PhoneGap é formado por várias API que permitem aceder às funções nativas dos dispositivos como a câmara, agenda, contactos, entre outras, através das linguagens JavaScript, HTML5 e CSS3, não sendo necessário utilizar linguagens específicas para os dispositivos como Objective-C. O desenvolvimento aplicacional assemelha-se ao de um site normal.

Tem como desvantagens os seguintes pontos:

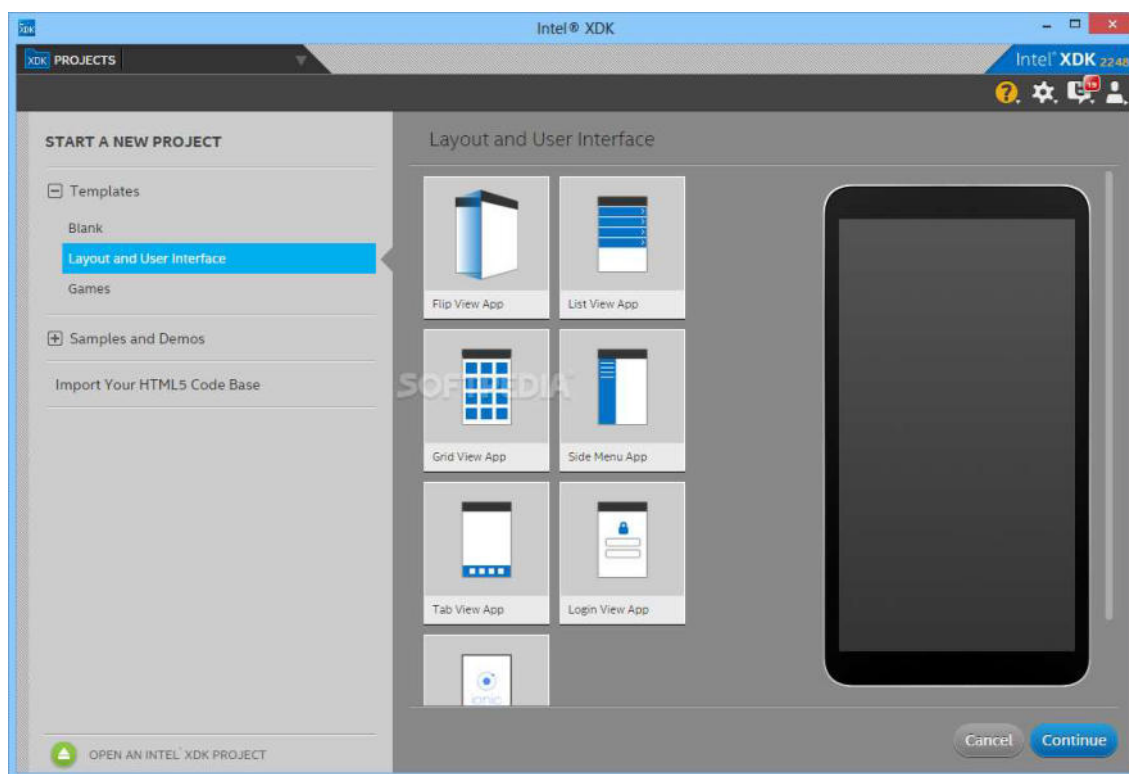
- fraco desempenho no caso de uso intensivo de gráficos.

- não disponibiliza elementos de raiz como *UI Widgets* ou comandos *standard*.
- Por vezes torna-se ineficiente quando utiliza aplicações nativas.
- A *framework PhoneGap* que programadores desenvolvam as aplicações apenas uma vez, a partir daí são cobradas taxa mensais (Malhotra, 2015).

8.4. Intel XDK

Outra ferramenta que se destacou foi o XDK da Intel. Apesar da Intel ser conhecida maioritariamente pelo desenvolvimento de *hardware* (Bloomberg, 2020), nomeadamente CPUs, controladores de rede, *chips* gráficos e *chipsets*. A Intel disponibiliza um conjunto de programas de demonstração e bibliografia no seu site para a plataforma XDK. As linguagens de programação utilizadas são dominadas pelo javascript e pelo html5. O XDK permite, de uma forma bastante simples, disponibilizar a mesma aplicação para qualquer plataforma seja IOS, *Android*, *WindowsPhone*, ou mesmo para *Windows*.

O XDK possui uma interface bastante agradável e intuitiva, o que facilita o desenvolvimento aplicacional, conforme demonstra a Figura 47 - Interface XDKFigura 47.

**Figura 47 - Interface XDK**

A ferramenta selecionada para o desenvolvimento do *smartCPE* foi o XDK. O *smartCPE* cumpre o objetivo proposto pois pode ser utilizado pelas plataformas IOS, *Android* e *WindowsPhone*. A estrutura, nesta fase, é composta por um *web service* efetua consultas e retorna informação da base de dados da Atena T, que é apresentada aos utilizadores finais, que são as equipas técnicas. Esta informação é apresentada sob a forma de listagem e refere-se às ordens de serviços de instalações de ADSL que deverão efetuar.

A distribuição da aplicação para os utilizadores é feita diretamente a partir do XDK: os programadores selecionam a plataforma de destino e a aplicação é enviada por email para o utilizador. Este faz a instalação através do link que consta no email, conforme indicado na Figura 48.

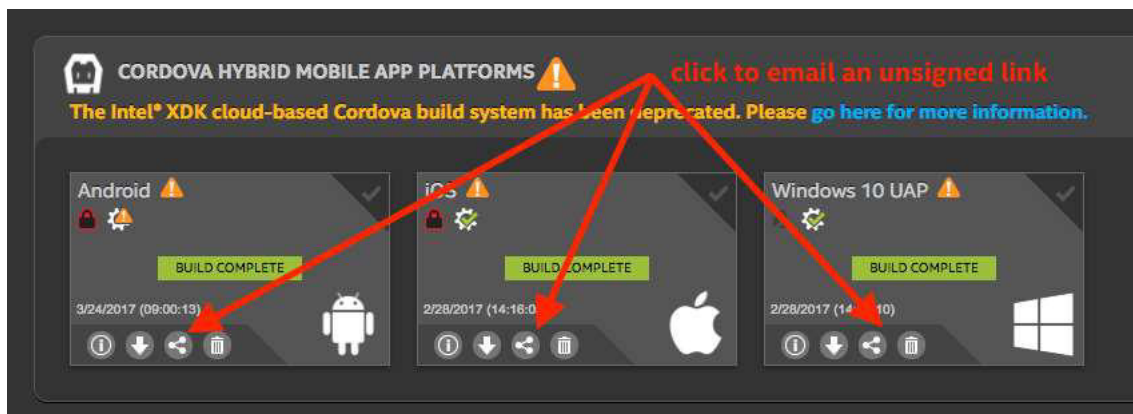


Figura 48 - XDK Link to email

O XDK acabou por ser descontinuado pela Intel em maio de 2017. O XDK para ser utilizado exige que se efetue login com as credenciais de acesso. Com o descontinuar do XDK tornou-se mandatório procurar uma nova plataforma de desenvolvimento para o *smartCPE*.

Durante o decorrer deste projeto verificou-se que, por uma questão operacional, os técnicos não utilizavam a aplicação móvel porque estavam habituados a utilizar o portátil para a configuração dos equipamentos e rede do cliente, e acabam por fazer o trabalho através do *webCPE*. Sendo assim a solução encontrada para que os técnicos possam ter acesso às funcionalidades do *smartCPE* nos dispositivos móveis foi adicionar a técnica de *responsive website* com *bootstrap* ao *webCPE*. Técnica que é descrita neste capítulo no ponto seguinte.

8.5. Responsive Website com bootstrap

Após verificar que a aplicação móvel não estava a ser utilizada e que o *website* não estava preparado para se obter uma visualização adequada em dispositivos móveis, chegou a altura de criar condições para uniformizar o *website* para todas as plataformas. A solução passa por adaptar o *website* para ser visualizado em qualquer ecrã, independente da sua dimensão, desde o *smartphone* até ao *desktop* (TutorialRepublic, 2020). A técnica utilizada é o *responsive web design*, e tem as seguintes características:

- Adapta o *layout* da página web à resolução do ecrã do dispositivo utilizado.

- Faz o redimensionamento das imagens automaticamente, não sobrecarregando a transferência de dados.
- Em dispositivos moveis oculta e adapta elementos desnecessários, como o caso de botões e links.

Surge então a oportunidade de fazer um *upgrade* ao *website* existente e recorre-se ao Bootstrap, que é uma *framework* de *Front End*, que permite a criação de *website responsive*. Para integrar esta tecnologia no webCPE houve necessidade de importar para o projeto *website* do webCPE os ficheiros “bootstrap.min.css” e “bootstrap.min.js”, que foram obtidos em <http://getbootstrap.com>. Assim já existem as condições necessárias para adaptar o *webCPE* para que possa ser acedido por qualquer tipo de dispositivo desde o portátil ao smartphone passando pelos tablets que permite então ter uma visualização homogénea. Os elementos da *framework* Bootstrap são simples de utilizar, pois estamos perante HTML5 e Javascript.

9. CONCLUSÃO

O desenvolvimento de *software* reveste-se de um nível elevado de complexidade. Quando este projeto foi inicialmente apresentado, a ideia transmitida seria para desenvolver uma pequena aplicação para interagir com um *web service* externo. Com o desenrolar do projeto foram surgindo necessidades e exigências que obrigaram a criação de um conjunto de aplicações. Como primeiro desafio a elaboração de uma pequena aplicação em JAVA SE, que seguidamente foi evoluida para um *website* com ligação ao sistema de ERP PHC, não esquecendo a vertente móvel.

Uma fase bastante importante foi a análise de mercado, porque foi decisiva para as opções tomadas. As aplicações desenvolvidas, *deskCPE*, *webCPE* e *smartCPE*, por recursos internos da Atena T permitiu atingir o objetivo pretendido com sucesso, conforme descrito nas diversas fases de desenvolvimento.

Quando chegou o momento de efetuar os testes do *deskCPE*, tudo decorreu com normalidade e houve validação por parte da Vodafone para entrada em produção.

A escolha das ferramentas de desenvolvimento é um fator que pode pesar no resultado final, como se verificou com escolha da ferramenta Intel XDK que acabou por ser colocada de lado em favor da tecnologia *responsive* com bootstrap. Nesta fase, é consumido bastante tempo, o que pode colocar em risco o cumprimento de prazos. Pelo que a tomada de decisões deve ser rápida.

Verifica-se que um projeto de desenvolvimento de *software* nunca está terminado e existe sempre algo a acrescentar. Quer seja devido a novas necessidades do negócio da Atena T com a Vodafone, alterações de design ou otimização do código fonte.

Verifica-se que existe necessidade de uma atenção constante para as novas tecnologias que possam ser usadas no desenvolvimento aplicacional, bem como para as novas necessidades que surjam para otimizar os recursos humanos e técnicos disponíveis, para assim dimensionar de acordo com os objetivos definidos pela empresa, nomeadamente os de ordem financeira.

O objetivo proposto pela administração da Atena T neste projeto foi concretizado e concluído com êxito.

Este projeto permitiu aplicar diretamente os conhecimentos obtidos a nível académico numa situação de desenvolvimento aplicacional em ambiente real.

Pode ainda concluir-se que um projeto desta dimensão deveria estar suportado numa equipa multidisciplinar para existir um foco específico, sobretudo um grupo para desenvolvimento do *webCPE* e um elemento para o desenvolvimento da parte móvel da aplicação. Assim poderia existir uma análise mais aprofundada em cada fase de desenvolvimento e implementação.

O expectável é que este projeto não termine com o desenvolvimento aqui finalizado, este poderá ser uma base de trabalho para uma evolução do *webCPE*.

Referências Bibliográficas

- Angelini, E. (28 de fevereiro de 2012). *enricoangelini*. Obtido de enricoangelini: <https://enricoangelini.com/2012/5-pros-and-cons-of-appcelerators-titanium/>
- Chang, R. (22 de janeiro de 2020). *tuts+*. Obtido de *tuts+*: <https://code.tutsplus.com/articles/mobile-operating-systems-in-2014--cms-19845>
- Cordova, A. (11 de outubro de 2019). *Apache Cordova*. Obtido de Apache Cordova: <https://cordova.apache.org/>
- Globalsign. (02 de setembro de 2019). *Globalsign*. Obtido de Globalsign: <https://www.globalsign.com/en/blog/the-difference-between-http-and-https/>
- hatchapps.com*. (02 de janeiro de 2019). Obtido de <https://hatchapps.com/blog/how-much-does-it-cost-to-build-an-app-this-study-shows-just-how-variable-pricing-really-is/>: <https://hatchapps.com/blog/how-much-does-it-cost-to-build-an-app-this-study-shows-just-how-variable-pricing-really-is/>
- http://docs.oasis-open.org*. (26 de junho de 2015). Obtido de <http://docs.oasis-open.org>: <http://docs.oasis-open.org/ubl/cs-UBL-2.0/uml/UBL-2.0-ApplicationResponseDocumentAssembly.html>
- http://docs.oasis-open.org/ubl/prd3-UBL-2.0/uml/UBL-2.0-ReceiptAdviceDocumentAssembly.html*. (26 de junho de 2015). Obtido de <http://docs.oasis-open.org/ubl/prd3-UBL-2.0/uml/UBL-2.0-ReceiptAdviceDocumentAssembly.html>
- http://www.infoempresas.com.pt*. (2016 de Março de 2016). Obtido de *infoempresas*: http://www.infoempresas.com.pt/J62_CONSULTORIA-PROGRAMACAO-INFORMATICA-ATIVIDADES-RELACIONADAS.html
- Investopedia*. (22 de abril de 2019). Obtido de *Investopedia*: <https://www.investopedia.com/articles/markets/030216/worlds-top-10-telecommunications-companies.asp>
- Luis Corral, A. J. (2012). Potential advantages and disadvantages of multiplatform. *Potential advantages and disadvantages of multiplatform*, p. 6.
- Microsoft. (10 de dezembro de 2019). *ASP.Net Blog*. Obtido de *DevBlogs*: <https://devblogs.microsoft.com/aspnet/accessing-the-asp-net-web-configuration-tool-in-visual-studio-2013/>
- Nemes, A. (18 de novembro de 2013). *Dotnetcodr*. Obtido de <http://dotnetcodr.com/2013/11/18/how-to-protect-your-config-file-in-net-with-cryptography/>: <http://dotnetcodr.com/2013/11/18/how-to-protect-your-config-file-in-net-with-cryptography/>
- Oracle. (02 de setembro de 2019). *Class Authenticator*. Obtido de Docs Oracle: <https://docs.oracle.com/javase/7/docs/api/java/net/Authenticator.html>
- Oracle. (02 de setembro de 2019). *Class PasswordAuthentication*. Obtido de Docs Oracle: <https://docs.oracle.com/javase/7/docs/api/java/net/PasswordAuthentication.html>

- Oracle. (02 de setembro de 2019). *Class X509TM*. Obtido de Docs Oracle: <https://docs.oracle.com/javase/7/docs/api/javax/net/ssl/X509TrustManager.html>
- Oracle. (02 de setembro de 2019). *Docs Oracle*. Obtido de Docs Oracle: <https://docs.oracle.com/cd/E19683-01/816-7264/6md9iem0s/index.html>
- Sroka, A. (19 de abril de 2016). *diwebsity*. Obtido de diwebsity: <https://www.diwebsity.com/2016/04/19/express-profiler/>
- UBL 2.1. (26 de julho de 2019). Obtido de UBL 2.1: <http://docs.oasis-open.org/ubl/UBL-2.1.html>
- UBL. (01 de janeiro de 2016). Obtido de UBL: <http://ubl.xml.org/>
- UBL Docs. (14 de junho de 2019). Obtido de UBL 2.1: <http://docs.oasis-open.org/ubl/os-UBL-2.1/UBL-2.1.html>
- w3schools. (02 de setembro de 2019). *w3schools*. Obtido de w3schools: https://www.w3schools.com/whatis/whatis_http.asp
- Wiki UBL. (14 de junho de 2019). Obtido de Wikipedia: https://en.wikipedia.org/wiki/Universal_Business_Language
- WSDL. (05 de agosto de 2019). Obtido de WSDL: [https://docs.microsoft.com/en-us/previous-versions/dotnet/netframework-2.0/7h3ystb6\(v=vs.80\)](https://docs.microsoft.com/en-us/previous-versions/dotnet/netframework-2.0/7h3ystb6(v=vs.80))

Anexo A – Classes Desenvolvidas

Este anexo tem o código de todas as classes desenvolvidas em JAVA no âmbito deste projeto.

ApplicationResponseBuil.java

```
package VodApp;

import java.util.ArrayList;
import java.util.List;
import oasis.names.specification.ubl.schema.xsd.applicationresponse_2.ApplicationResponseType;
import oasis.names.specification.ubl.schema.xsd.commonaggregatecomponents_2.DocumentResponseType;
import oasis.names.specification.ubl.schema.xsd.commonaggregatecomponents_2.ResponseType;
import oasis.names.specification.ubl.schema.xsd.commonbasiccomponents_2.DescriptionType;
import oasis.names.specification.ubl.schema.xsd.commonbasiccomponents_2.ReferenceIDType;
import oasis.names.specification.ubl.schema.xsd.commonbasiccomponents_2.ResponseCodeType;

public class ApplicationResponseBuild {

    private static ApplicationResponseType appResposta = new ApplicationResponseType();

    public static ApplicationResponseType appResponse (){

        List<DescriptionType> listDescricao = new ArrayList<>();
        DescriptionType descricao = new DescriptionType();
        descricao.setValue("Processing");
        listDescricao.add(descricao);

        ResponseType respType = new ResponseType();
        respType.getDescription().add(descricao);
        ResponseCodeType respCode = new ResponseCodeType();
        respCode.setValue("11111");
        respType.setResponseCode(respCode);
        ReferenceIDType refId = new ReferenceIDType();
        refId.setValue("FS10001010");
        respType.setReferenceID(refId);
        DocumentResponseType docResponse = new DocumentResponseType();
        docResponse.setResponse(respType);
        appResposta.getDocumentResponse().add(docResponse);

        return appResposta;
    }
}
```

BuyCustParty.java

```

package VodApp;

import java.util.ArrayList;
import java.util.List;
import
oasis.names.specification.ubl.schema.xsd.commonaggregatecomponents_2.CustomerPartyType;
import
oasis.names.specification.ubl.schema.xsd.commonaggregatecomponents_2.PartyIdentificationType;
import oasis.names.specification.ubl.schema.xsd.commonaggregatecomponents_2.PartyNameType;
import
oasis.names.specification.ubl.schema.xsd.commonaggregatecomponents_2.PartyTaxSchemeType;
import oasis.names.specification.ubl.schema.xsd.commonaggregatecomponents_2.PartyType;
import
oasis.names.specification.ubl.schema.xsd.commonaggregatecomponents_2.SupplierPartyType;
import oasis.names.specification.ubl.schema.xsd.commonaggregatecomponents_2.TaxSchemeType;
import
oasis.names.specification.ubl.schema.xsd.commonbasiccomponents_2.AdditionalAccountIDType;
import oasis.names.specification.ubl.schema.xsd.commonbasiccomponents_2.CompanyIDType;
import oasis.names.specification.ubl.schema.xsd.commonbasiccomponents_2.IDType;
import oasis.names.specification.ubl.schema.xsd.commonbasiccomponents_2.NameType;
import
oasis.names.specification.ubl.schema.xsd.commonbasiccomponents_2.SupplierAssignedAccountIDTy
pe;
import oasis.names.specification.ubl.schema.xsd.commonbasiccomponents_2.TaxTypeCodeType;

public class BuyCustParty {

    public static CustomerPartyType custBCP ( String supAccId, String accessid, String
name, String compId){

        IDType idTax = new IDType();
        TaxTypeCodeType taxCode = new TaxTypeCodeType();
        CompanyIDType companyId = new CompanyIDType();
        TaxSchemeType taxScheme = new TaxSchemeType();
        PartyTaxSchemeType PartyTaxScheme = new PartyTaxSchemeType();
        List<PartyTaxSchemeType> partyTaxScheme = new ArrayList<>();
        PartyType PartyType = new PartyType();
        List<PartyNameType> partyName = new ArrayList<PartyNameType>();
        PartyNameType PartyName = new PartyNameType();
        NameType nameType = new NameType();
        SupplierAssignedAccountIDType supplierAssignedAccountID = new
SupplierAssignedAccountIDType();

        List<AdditionalAccountIDType> additionalAccountID = new ArrayList<>();
        AdditionalAccountIDType AccessId = new AdditionalAccountIDType();
        CustomerPartyType BuyerCustomerParty = new CustomerPartyType();
        idTax.setValue("VAT");
        taxCode.setValue("VAT");
    }
}

```

```
taxScheme.setID(idTax);
taxScheme.setTaxTypeCode(taxCode);
companyId.setValue(compId);
PartyTaxScheme.setCompanyID(companyId);
PartyTaxScheme.setTaxScheme(taxScheme);
if (name != null){
    nameType.setValue(name);
    PartyName.setName(nameType);
}
AccessId.setValue(accessid);
    supplierAssignedAccountID.setValue(supAccId);
PartyType.getPartyName().add(PartyName);
PartyType.getPartyTaxScheme().add(PartyTaxScheme);
BuyerCustomerParty.setParty(PartyType);
BuyerCustomerParty.setSupplierAssignedAccountID(supplierAssignedAccountID);
BuyerCustomerParty.getAdditionalAccountID().add(AccessId);
return BuyerCustomerParty;
}
}
```

ClearReceiptAdvice.java

```
package VodApp;

import
oasis.names.specification.ubl.schema.xsd.commonaggregatecomponents_2.CustomerPartyType;
import
oasis.names.specification.ubl.schema.xsd.commonaggregatecomponents_2.SupplierPartyType;
import oasis.names.specification.ubl.schema.xsd.receiptadvice_2.ReceiptAdviceType;

public class ClearReceiptAdvice {

    private static CustomerPartyType BuyerCustomerParty = new CustomerPartyType();
    private static CustomerPartyType DeliveryCustParty = new CustomerPartyType();
    private static SupplierPartyType DespatchSupplierParty = new SupplierPartyType();
    private static ReceiptAdviceType recAdvice = new ReceiptAdviceType();

    public static ReceiptAdviceType clearRecAdvice (ReceiptAdviceType clearRAdvice){

        clearRAdvice = recAdvice;

        return recAdvice;
    }
}
```

DelCustParty.java

```
package VodApp;

import java.util.ArrayList;
import java.util.List;
import oasis.names.specification.ubl.schema.xsd.commonaggregatecomponents_2.CustomerPartyType;
import oasis.names.specification.ubl.schema.xsd.commonaggregatecomponents_2.PartyIdentificationType;
import oasis.names.specification.ubl.schema.xsd.commonaggregatecomponents_2.PartyNameType;
import oasis.names.specification.ubl.schema.xsd.commonaggregatecomponents_2.PartyTaxSchemeType;
import oasis.names.specification.ubl.schema.xsd.commonaggregatecomponents_2.PartyType;
import oasis.names.specification.ubl.schema.xsd.commonaggregatecomponents_2.SupplierPartyType;
import oasis.names.specification.ubl.schema.xsd.commonaggregatecomponents_2.TaxSchemeType;
import oasis.names.specification.ubl.schema.xsd.commonbasiccomponents_2.AdditionalAccountIDType;
import oasis.names.specification.ubl.schema.xsd.commonbasiccomponents_2.CompanyIDType;
import oasis.names.specification.ubl.schema.xsd.commonbasiccomponents_2.IDType;
import oasis.names.specification.ubl.schema.xsd.commonbasiccomponents_2.NameType;
import oasis.names.specification.ubl.schema.xsd.commonbasiccomponents_2.TaxTypeCodeType;

public class DelCustParty {
    public static CustomerPartyType custDCP (String idDcp, String name, String compId,String
    idvat ){

        IDType idTax = new IDType();
        TaxTypeCodeType taxCode = new TaxTypeCodeType();
        CompanyIDType companyId = new CompanyIDType();
        TaxSchemeType taxScheme = new TaxSchemeType();
        PartyTaxSchemeType PartyTaxScheme = new PartyTaxSchemeType();
        PartyType PartyType = new PartyType();
        List<PartyNameType> partyName = new ArrayList<PartyNameType>();
        PartyNameType PartyName = new PartyNameType();
        NameType nameType = new NameType();

        List<PartyTaxSchemeType> partyTaxScheme = new ArrayList<>();
        CustomerPartyType DeliveryCustParty = new CustomerPartyType();
        List<PartyIdentificationType> ListPartyId = new ArrayList<>();
        PartyIdentificationType PartyId = new PartyIdentificationType();
        IDType id = new IDType();
        idTax.setValue(idvat);
        taxCode.setValue("VAT");
        taxScheme.setID(idTax);
        taxScheme.setTaxTypeCode(taxCode);
```



```
        companyId.setValue(compId);
        PartyTaxScheme.setCompanyID(companyId);
        PartyTaxScheme.setTaxScheme(taxScheme);
        partyTaxScheme.add(0,PartyTaxScheme);
        nameType.setValue(name);
        PartyName.setName(nameType);
        id.setValue(idDcp);
        PartyId.setID(id);
        ListPartyId.add(0, PartyId);
        partyName.add(0,PartyName);
        PartyType.getPartyIdentification().add(PartyId);
        PartyType.getPartyName().add(PartyName);
        PartyType.getPartyTaxScheme().add(PartyTaxScheme);
        DeliveryCustParty.setParty(PartyType);
    return DeliveryCustParty;
}
}
```

DespSupParty.java

```

package VodApp;

import java.util.ArrayList;
import java.util.List;
import oasis.names.specification.ubl.schema.xsd.commonaggregatecomponents_2.CustomerPartyType;
import oasis.names.specification.ubl.schema.xsd.commonaggregatecomponents_2.PartyIdentificationType;
import oasis.names.specification.ubl.schema.xsd.commonaggregatecomponents_2.PartyNameType;
import oasis.names.specification.ubl.schema.xsd.commonaggregatecomponents_2.PartyTaxSchemeType;
import oasis.names.specification.ubl.schema.xsd.commonaggregatecomponents_2.PartyType;
import oasis.names.specification.ubl.schema.xsd.commonaggregatecomponents_2.SupplierPartyType;
import oasis.names.specification.ubl.schema.xsd.commonaggregatecomponents_2.TaxSchemeType;
import oasis.names.specification.ubl.schema.xsd.commonbasiccomponents_2.AdditionalAccountIDType;
import oasis.names.specification.ubl.schema.xsd.commonbasiccomponents_2.CompanyIDType;
import oasis.names.specification.ubl.schema.xsd.commonbasiccomponents_2.IDType;
import oasis.names.specification.ubl.schema.xsd.commonbasiccomponents_2.NameType;
import oasis.names.specification.ubl.schema.xsd.commonbasiccomponents_2.TaxTypeCodeType;

public class DespSupParty {

    public static SupplierPartyType custDSP ( String idDsp, String name, String compId,
String idvat ){

        IDType idTax = new IDType();
        TaxTypeCodeType taxCode = new TaxTypeCodeType();
        CompanyIDType companyId = new CompanyIDType();
        TaxSchemeType taxScheme = new TaxSchemeType();
        PartyTaxSchemeType PartyTaxScheme = new PartyTaxSchemeType();
        List<PartyTaxSchemeType> partyTaxScheme = new ArrayList<>();
        PartyType PartyType = new PartyType();
        List<PartyNameType> partyName = new ArrayList<PartyNameType>();
        PartyNameType PartyName = new PartyNameType();
        NameType nameType = new NameType();
        SupplierPartyType DespatchSupplierParty = new SupplierPartyType();
        List<PartyIdentificationType> partyIdentification = new ArrayList<>();
        PartyIdentificationType PartyId = new PartyIdentificationType();
        IDType id = new IDType();
        idTax.setValue(idvat);
        taxCode.setValue("VAT");
        taxScheme.setID(idTax);
        taxScheme.setTaxTypeCode(taxCode);
        companyId.setValue(compId);
    }
}

```

```
PartyTaxScheme.setCompanyID(companyId);
PartyTaxScheme.setTaxScheme(taxScheme);
partyTaxScheme.add(0, PartyTaxScheme);
nameType.setValue(name);
PartyName.setName(nameType);
id.setValue(idDsp);
PartyId.setID(id);
partyIdentification.add(0, PartyId);
partyName.add(0, PartyName);
PartyType.getPartyIdentification().add(PartyId);
PartyType.getPartyName().add(PartyName);
PartyType.getPartyTaxScheme().add(PartyTaxScheme);
DespatchSupplierParty.setParty(PartyType);

return DespatchSupplierParty;
}
}
```

DocReference.java

```
package VodApp;

import
    oasis.names.specification.ubl.schema.xsd.commonaggregatecomponents_2.DocumentReferenceType;
import oasis.names.specification.ubl.schema.xsd.commonbasiccomponents_2.IDType;

public class DocReference {
    public static DocumentReferenceType DocRef ( String DocRefId){
        DocumentReferenceType documentReference = new DocumentReferenceType();
        IDType docreferenceId = new IDType();
        docreferenceId.setValue(DocRefId);
        documentReference.setID(docreferenceId);

        return documentReference;
    }
}
```

RecLineList.java

```

package VodApp;

import java.math.BigDecimal;
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;
import oasis.names.specification.ubl.schema.xsd.commonaggregatecomponents_2.ItemIdentificationType;
import oasis.names.specification.ubl.schema.xsd.commonaggregatecomponents_2.ItemInstanceType;
import oasis.names.specification.ubl.schema.xsd.commonaggregatecomponents_2.ItemType;
import oasis.names.specification.ubl.schema.xsd.commonaggregatecomponents_2.ReceiptLineType;
import oasis.names.specification.ubl.schema.xsd.commonbasiccomponents_2.IDType;
import oasis.names.specification.ubl.schema.xsd.commonbasiccomponents_2.ReceivedQuantityType;
import oasis.names.specification.ubl.schema.xsd.commonbasiccomponents_2.RejectedQuantityType;
import oasis.names.specification.ubl.schema.xsd.commonbasiccomponents_2.SerialIDType;

public class RecLineList {

    private static ReceiptLineType receiptLine1 = new ReceiptLineType();

    private static List<ReceiptLineType> receiptLineList = new ArrayList<>();

    public static List<ReceiptLineType> ReceiptLineList(List<String> itemID,
List<String> Receivedqt,
        List<String> Rejectedqt, List<String> Material, List<String> Serie) {

        for (int i = 0; i < itemID.size(); i++) {
            receiptLine1 = ReceiptLine1(itemID.get(i), Receivedqt.get(i), Rejectedqt.get(i),
Material.get(i), Serie.get(i));
            receiptLineList.add(i, receiptLine1);
        }

        return Collections.unmodifiableList(receiptLineList);
    }

    private static ReceiptLineType ReceiptLine1 (String idline, String received, String
rejected, String seller, String serie){

        ReceiptLineType receiptLine = new ReceiptLineType();
        IDType SellerId = new IDType();
        ItemType item1 = new ItemType();
        IDType idRecLine = new IDType();
        SerialIDType serialId = new SerialIDType();
        idRecLine.setValue(idline);
        receiptLine.setID(idRecLine);

        serialId.setValue(serie);
        ItemInstanceType itemInstance = new ItemInstanceType();
        itemInstance.setSerialID(serialId);
    }
}

```

```
        SellerId.setValue(seller);
        ItemIdentificationType sellersItemId = new ItemIdentificationType();
        sellersItemId.setID(SellerId);

        item1.setSellersItemIdentification(sellersItemId);
        item1.getItemInstance().add(itemInstance);

        if (received != null){
            ReceivedQuantityType receivedQuantity = new ReceivedQuantityType();
            BigDecimal receivedBig = new BigDecimal(received);
            receivedQuantity.setValue(receivedBig);
            receiptLine.setReceivedQuantity(receivedQuantity);
        }

        if (rejected != null) {
            RejectedQuantityType rejectedQuantity = new RejectedQuantityType();
            BigDecimal rejectedBig = new BigDecimal(rejected);
            rejectedQuantity.setValue(rejectedBig);
            receiptLine.setRejectedQuantity(rejectedQuantity);
        }

        receiptLine.getItem().add(item1);
        return receiptLine;
    }
}
```

RespostaBuild.java

```
package VodApp;

import java.util.ArrayList;
import java.util.List;
import oasis.names.specification.ubl.schema.xsd.commonaggregatecomponents_2.ResponseType;
import oasis.names.specification.ubl.schema.xsd.commonbasiccomponents_2.DescriptionType;
import oasis.names.specification.ubl.schema.xsd.commonbasiccomponents_2.ReferenceIDType;
import oasis.names.specification.ubl.schema.xsd.commonbasiccomponents_2.ResponseCodeType;

public class RespostaBuild {

    private static ResponseType respType = new ResponseType();

    public static ResponseType response () {

        List<DescriptionType> listDescricao = new ArrayList<>();
        DescriptionType descricao = new DescriptionType();
        descricao.setValue("Processing");
        listDescricao.add(descricao);

        respType.getDescription().add(descricao);
        ResponseCodeType respCode = new ResponseCodeType();
        respCode.setValue("11111");

        respType.setResponseCode(respCode);
        ReferenceIDType refId = new ReferenceIDType();
        refId.setValue("FS10001010");
        respType.setReferenceID(refId);

        return respType;
    }
}
```

RespostaCheckStatus.java

```
package VodApp;

import
oasis.names.specification.ubl.schema.xsd.applicationresponse_2.ApplicationResponseType;
import
oasis.names.specification.ubl.schema.xsd.commonaggregatecomponents_2.DocumentResponseType;
import oasis.names.specification.ubl.schema.xsd.commonaggregatecomponents_2.ResponseType;

public class RespostaCheckStatus {

    private static String IdResposta = new String();
    private static DocumentResponseType respostaType = new DocumentResponseType();
    private static String CodResposta = new String();
    private static String DescricaoResposta = new String();

    public static String DocRef ( ApplicationResponseType Resposta){
        respostaType = Resposta.getDocumentResponse().get(0);
        IdResposta = respostaType.getResponse().getReferenceID().getValue();
        return IdResposta;
    }

    public static String CodRef ( ApplicationResponseType Resposta){
        respostaType = Resposta.getDocumentResponse().get(0);
        CodResposta = respostaType.getResponse().getResponseCode().getValue();
        return CodResposta;
    }

    public static String DescricaoRef ( ApplicationResponseType Resposta){
        respostaType = Resposta.getDocumentResponse().get(0);
        DescricaoResposta = respostaType.getResponse().getDescription().get(0).getValue();
        return DescricaoResposta;
    }

}
```


RespostaConsign.java

```
package VodApp;

import oasis.names.specification.ubl.schema.xsd.commonaggregatecomponents_2.ResponseType;

public class RespostaConsign {

    private static String IdResposta = new String();
    private static String CodResposta = new String();
    private static String DescricaoResposta = new String();

    public static String DocRef ( ResponseType Resposta){
        IdResposta = Resposta.getReferenceID().getValue();
        return IdResposta;
    }

    public static String CodRef ( ResponseType Resposta){
        CodResposta = Resposta.getResponseCode().getValue();
        return CodResposta;
    }

    public static String DescricaoRef ( ResponseType Resposta){
        DescricaoResposta = Resposta.getDescription().get(0).getValue();
        return DescricaoResposta;
    }
}
```

JFrameNoBd.java

```
/

package VodApp;

import java.io.StringWriter;
import java.net.Authenticator;
import java.net.PasswordAuthentication;
import java.util.ArrayList;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.net.ssl.HttpURLConnection;
import javax.net.ssl.SSLContext;
import javax.net.ssl.TrustManager;
import javax.net.ssl.X509TrustManager;
import javax.swing.JInternalFrame;
import javax.swing.table.DefaultTableModel;
import javax.xml.bind.JAXBContext;
import javax.xml.bind.JAXBElement;
import javax.xml.bind.JAXBException;
import javax.xml.bind.Marshaller;
import javax.xml.datatype.DatatypeConfigurationException;
import javax.xml.datatype.DatatypeFactory;
import javax.xml.datatype.XMLGregorianCalendar;
import
oasis.names.specification.ubl.schema.xsd.applicationresponse_2.ApplicationResponseType;
import
oasis.names.specification.ubl.schema.xsd.commonaggregatecomponents_2.CustomerPartyType;
import
oasis.names.specification.ubl.schema.xsd.commonaggregatecomponents_2.DocumentReferenceType;
import oasis.names.specification.ubl.schema.xsd.commonaggregatecomponents_2.ResponseType;
import
oasis.names.specification.ubl.schema.xsd.commonaggregatecomponents_2.SupplierPartyType;
import oasis.names.specification.ubl.schema.xsd.commonbasiccomponents_2.CopyIndicatorType;
import oasis.names.specification.ubl.schema.xsd.commonbasiccomponents_2.IDType;
import oasis.names.specification.ubl.schema.xsd.commonbasiccomponents_2.IssueDateType;
import oasis.names.specification.ubl.schema.xsd.commonbasiccomponents_2.UBLVersionIDType;
import oasis.names.specification.ubl.schema.xsd.commonbasiccomponents_2.UUIDType;
import oasis.names.specification.ubl.schema.xsd.receiptadvice_2.ObjectFactory;
import oasis.names.specification.ubl.schema.xsd.receiptadvice_2.ReceiptAdviceType;
import pt.vodafone.consignstock.ConsignStock;

/**
 *
 * @author paulom
 */
public class JFrameNoBd extends javax.swing.JFrame{
```

```

/**
 * Creates new form JFrameNoBd
 */
private static List <String> itemID = new ArrayList<String>();
private static List <String> Receivedqt = new ArrayList<>();
private static List <String> Rejectedqt = new ArrayList<>();
private static List <String> Material = new ArrayList<>();
private static List <String> Serie = new ArrayList<>();
// private static ReceiptAdviceType receiptAdvice = new ReceiptAdviceType();
private static UBLVersionIDType UBL = new UBLVersionIDType(); //sempre 2.0
private static IDType idWorkorder = new IDType();
private static CopyIndicatorType copyIndicator = new CopyIndicatorType(); // sempre
private static IssueDateType issueDate = new IssueDateType();
private static UUIDType uuid = new UUIDType();
private static ReceiptAdviceType clearReceiptAdvice = new ReceiptAdviceType();

public JFrameNoBd() {
    initComponents();

    //jPanelConsign.setVisible(false);
    jInternalFrameWorkorder.setVisible(false);
    jInternalFrameCheckStatus.setVisible(false);

}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    menuBar1 = new java.awt.MenuBar();
    menu1 = new java.awt.Menu();
    menu2 = new java.awt.Menu();
    jInternalFrameWorkorder = new javax.swing.JInternalFrame();
    jPanelConsign = new javax.swing.JPanel();
    jLabelContaCliente = new javax.swing.JLabel();
    jTextFieldContaCliente = new javax.swing.JTextField();
    jLabelAccessId = new javax.swing.JLabel();
    jTextFieldAccessId = new javax.swing.JTextField();
    jLabelNomeCliente = new javax.swing.JLabel();
    jTextFieldNomeCliente = new javax.swing.JTextField();
    jLabelWorkOrder = new javax.swing.JLabel();
    jTextFieldWorkOrder = new javax.swing.JTextField();
    jLabel11 = new javax.swing.JLabel();
    jTextFieldContribuinteCliente = new javax.swing.JTextField();
    jScrollPane1 = new javax.swing.JScrollPane();
    jTableItens = new javax.swing.JTable();
    jLabelTitleWo = new javax.swing.JLabel();

```

```
jButtonEnviar = new javax.swing.JButton();
jScrollPane2 = new javax.swing.JScrollPane();
jTextAreaReceiptAdvice = new javax.swing.JTextArea();
jLabelData = new javax.swing.JLabel();
jTextFieldData = new javax.swing.JTextField();
jLabeldataexemplo = new javax.swing.JLabel();
jLabelFsId = new javax.swing.JLabel();
jTextFieldFsId = new javax.swing.JTextField();
jLabelUUID = new javax.swing.JLabel();
jTextFieldUUID = new javax.swing.JTextField();
jPanelRespostaConsign = new javax.swing.JPanel();
jLabelTitleResposta = new javax.swing.JLabel();
jLabelRefId = new javax.swing.JLabel();
jLabelCodigo = new javax.swing.JLabel();
jLabelDescricao = new javax.swing.JLabel();
jTextFieldRefId = new javax.swing.JTextField();
jTextFieldCodigo = new javax.swing.JTextField();
jTextFieldDescricao = new javax.swing.JTextField();
jInternalFrameCheckStatus = new javax.swing.JInternalFrame();
jLabelTitleCheckStatus = new javax.swing.JLabel();
jLabelWorkOrder1 = new javax.swing.JLabel();
jTextFieldWorkOrder1 = new javax.swing.JTextField();
jButtonCheckS = new javax.swing.JButton();
jScrollPane3 = new javax.swing.JScrollPane();
jTextArea1 = new javax.swing.JTextArea();
jPanelRespostaConsign1 = new javax.swing.JPanel();
jLabelTitleResposta1 = new javax.swing.JLabel();
jLabelRefIdCS = new javax.swing.JLabel();
jLabelCodigoCS = new javax.swing.JLabel();
jLabelDescricaoCS = new javax.swing.JLabel();
jTextFieldRefIdCS = new javax.swing.JTextField();
jTextFieldCodigoCS = new javax.swing.JTextField();
jTextFieldDescricaoCS = new javax.swing.JTextField();
jScrollPane4 = new javax.swing.JScrollPane();
jTextAreaDebug = new javax.swing.JTextArea();
jScrollPane5 = new javax.swing.JScrollPane();
jTextAreaDebug1 = new javax.swing.JTextArea();
jMenuBarNoBd = new javax.swing.JMenuBar();
jMenu1 = new javax.swing.JMenu();
jMenuItemConsign = new javax.swing.JMenuItem();
jMenuItemCheckStatus = new javax.swing.JMenuItem();
jMenu2 = new javax.swing.JMenu();

menu1.setLabel("File");
menuBar1.add(menu1);

menu2.setLabel("Edit");
menuBar1.add(menu2);
```

```

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setPreferredSize(new java.awt.Dimension(800, 1000));

jInternalFrameWorkorder.setClosable(true);
jInternalFrameWorkorder.setIconifiable(true);
jInternalFrameWorkorder.setResizable(true);
jInternalFrameWorkorder.setPreferredSize(new java.awt.Dimension(788, 1033));
jInternalFrameWorkorder.setVisible(true);
jInternalFrameWorkorder.addInternalFrameListener(new
javax.swing.event.InternalFrameListener() {
    public void internalFrameActivated(javax.swing.event.InternalFrameEvent evt) {
    }
    public void internalFrameClosed(javax.swing.event.InternalFrameEvent evt) {
        jInternalFrameWorkorderInternalFrameClosed(evt);
    }
    public void internalFrameClosing(javax.swing.event.InternalFrameEvent evt) {
    }
    public void internalFrameDeactivated(javax.swing.event.InternalFrameEvent evt)
{
    }
    public void internalFrameDeiconified(javax.swing.event.InternalFrameEvent evt)
{
    }
    public void internalFrameIconified(javax.swing.event.InternalFrameEvent evt) {
    }
    public void internalFrameOpened(javax.swing.event.InternalFrameEvent evt) {
    }
});

jPanelConsign.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelB
order.RAISED));
jPanelConsign.setAlignmentX(1.0F);
jPanelConsign.setEnabled(false);

jLabelContaCliente.setText("Conta Cliente");

jTextFieldContaCliente.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jTextFieldContaClienteActionPerformed(evt);
    }
});

jLabelAccessId.setText("ADSL Id");

jTextFieldAccessId.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jTextFieldAccessIdActionPerformed(evt);
    }
});

```

```

jLabelNomeCliente.setText("Nome Cliente");

jLabelWorkOrder.setText("Workorder n° ");

jLabel1.setText("Contribuinte n° ");

jTableItens.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null, null},
        {null, null, null, null, null},
        {null, null, null, null, null},
        {null, null, null, null, null},
        {null, null, null, null, null},
        {null, null, null, null, null},
        {null, null, null, null, null},
        {null, null, null, null, null},
        {null, null, null, null, null},
        {null, null, null, null, null}
    },
    new String [] {
        "Item ID", "Received", "Rejected", "Material", "N° série"
    }
));
jTableItens.setColumnSelectionAllowed(true);
jScrollPane.setViewportView(jTableItens);

jLabelTitleWo.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
jLabelTitleWo.setText("Workorder");

jButtonEnviar.setText("Enviar");
jButtonEnviar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButtonEnviarActionPerformed(evt);
    }
});

jTextAreaReceiptAdvice.setColumns(20);
jTextAreaReceiptAdvice.setRows(100);
jScrollPane2.setViewportView(jTextAreaReceiptAdvice);

jLabelData.setText("Data");

jLabeldataexemplo.setText("ex: 2013-10-22");

jLabelFsId.setText("FS ID");

jLabelUUID.setText("UUID");

javax.swing.GroupLayout jPanelConsignLayout = new
javax.swing.GroupLayout(jPanelConsign);

```

```

jPanelConsign.setLayout(jPanelConsignLayout);
jPanelConsignLayout.setHorizontalGroup(

jPanelConsignLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanelConsignLayout.createSequentialGroup()
        .addGap(19, 19, 19)

.addGroup(jPanelConsignLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jLabelTitleWo, javax.swing.GroupLayout.PREFERRED_SIZE,
180, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGroup(jPanelConsignLayout.createSequentialGroup()

.addGroup(jPanelConsignLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanelConsignLayout.createSequentialGroup()

.addGroup(jPanelConsignLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
    .addGroup(jPanelConsignLayout.createSequentialGroup()
        .addComponent(jLabelWorkOrder)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
    .addComponent(jTextFieldWorkOrder,
javax.swing.GroupLayout.PREFERRED_SIZE, 113, javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGroup(jPanelConsignLayout.createSequentialGroup()

.addGroup(jPanelConsignLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jLabelContaCliente)
    .addComponent(jLabelNomeCliente))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanelConsignLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
    .addComponent(jTextFieldNomeCliente,
javax.swing.GroupLayout.PREFERRED_SIZE, 519, javax.swing.GroupLayout.PREFERRED_SIZE)

.addGroup(jPanelConsignLayout.createSequentialGroup())

.addGroup(jPanelConsignLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanelConsignLayout.createSequentialGroup())

.addComponent(jTextFieldContaCliente, javax.swing.GroupLayout.PREFERRED_SIZE, 99,
javax.swing.GroupLayout.PREFERRED_SIZE)

    .addGap(26, 26, 26)
    .addComponent(jLabelAccessId))

.addGroup(jPanelConsignLayout.createSequentialGroup())
    .addGap(134, 134, 134)
    .addComponent(jLabelData)))
    .addGap(18, 18, 18)

.addGroup(jPanelConsignLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```

.addGroup(jPanelConsignLayout.createSequentialGroup())
                                .addComponent(jTextFieldAccessId,
javax.swing.GroupLayout.PREFERRED_SIZE, 122, javax.swing.GroupLayout.PREFERRED_SIZE)
                                .addGap(18, 18, 18)
                                .addComponent(jLabel11)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jTextFieldContribuinteCliente))

.addGroup(jPanelConsignLayout.createSequentialGroup())
                                .addComponent(jLabeldataexemplo)
                                .addGap(0, 0, Short.MAX_VALUE))

.addGroup(jPanelConsignLayout.createSequentialGroup())
                                .addComponent(jTextFieldData,
javax.swing.GroupLayout.PREFERRED_SIZE, 85, javax.swing.GroupLayout.PREFERRED_SIZE)
                                .addGap(18, 18, 18)
                                .addComponent(jLabelFsId)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                                .addComponent(jTextFieldFsId,
javax.swing.GroupLayout.PREFERRED_SIZE, 65, javax.swing.GroupLayout.PREFERRED_SIZE)
                                .addGap(18, 18, 18)
                                .addComponent(jLabelUUID)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                                .addComponent(jTextFieldUUID))))))
                                .addComponent(jScrollPane1))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 44, Short.MAX_VALUE))
                                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanelConsignLayout.createSequentialGroup()
                                .addGap(0, 0, Short.MAX_VALUE)
                                .addComponent(jButtonEnviar)
                                .addGap(66, 66, 66)))
                                .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE,
243, javax.swing.GroupLayout.PREFERRED_SIZE)))
                                .addGap(34, 34, 34))
);
jPanelConsignLayout.setVerticalGroup(

jPanelConsignLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                .addGroup(jPanelConsignLayout.createSequentialGroup()
                                .addGap(18, 18, 18)
                                .addComponent(jLabelTitleWo)
                                .addGap(18, 18, 18)

.addGroup(jPanelConsignLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```



```

        .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE, 231,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGroup(jPanelConsignLayout.createSequentialGroup())

.addGroup(jPanelConsignLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jLabelWorkOrder)
        .addComponent(jTextFieldWorkOrder,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabelData)
        .addComponent(jTextFieldData,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabelFsId)
        .addComponent(jTextFieldFsId,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabelUUID)
        .addComponent(jTextFieldUUID,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jLabeldataexemplo)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanelConsignLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jLabelContaCliente)
        .addComponent(jTextFieldContaCliente,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabelAccessId)
        .addComponent(jTextFieldAccessId,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel11)
        .addComponent(jTextFieldContribuinteCliente,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanelConsignLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jTextFieldNomeCliente,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabelNomeCliente))
        .addGap(18, 18, 18)
        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE,
100, javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(jButtonEnviar)))
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    );

jPanelRespostaConsign.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.bord
er.BevelBorder.RAISED));

jLabelTitleResposta.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
jLabelTitleResposta.setText("Resposta");

jLabelRefId.setText("Ref Id");

jLabelCodigo.setText("Código");

jLabelDescricao.setText("Descrição");

javax.swing.GroupLayout jPanelRespostaConsignLayout = new
javax.swing.GroupLayout(jPanelRespostaConsign);
jPanelRespostaConsign.setLayout(jPanelRespostaConsignLayout);
jPanelRespostaConsignLayout.setHorizontalGroup(

jPanelRespostaConsignLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanelRespostaConsignLayout.createSequentialGroup()

.addGroup(jPanelRespostaConsignLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)

    .addGroup(jPanelRespostaConsignLayout.createSequentialGroup()

.addGroup(jPanelRespostaConsignLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING, false)

        .addGroup(jPanelRespostaConsignLayout.createSequentialGroup()
            .addGroup(jPanelRespostaConsignLayout.createParallelGroup(javax.swing.GroupLayout.A
lignment.LEADING)
                .addGroup(jPanelRespostaConsignLayout.createSequentialGroup()
                    .addGap(16, 16, 16)
                    .addComponent(jTextFieldRefId,
javax.swing.GroupLayout.PREFERRED_SIZE, 108, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(28, 28, 28)
                    .addComponent(jTextFieldCodigo,
javax.swing.GroupLayout.PREFERRED_SIZE, 67, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(15, 15, 15))
                .addGroup(jPanelRespostaConsignLayout.createSequentialGroup()
                    .addGap(20, 20, 20)
                    .addComponent(jLabelRefId)

            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(jLabelCodigo)
            .addGap(36, 36, 36)))

        .addGroup(jPanelRespostaConsignLayout.createParallelGroup(javax.swing.GroupLayout.A
lignment.LEADING)

```

```

        .addGroup(jPanelRespostaConsignLayout.createSequentialGroup()
            .addGap(77, 77, 77)
            .addComponent(jLabelDescricao)
            .addGap(0, 486, Short.MAX_VALUE))
        .addComponent(jTextFieldDescricao)))
        .addGroup(jPanelRespostaConsignLayout.createSequentialGroup()
            .addGap(196, 196, 196)
            .addComponent(jLabelTitleResposta,
                javax.swing.GroupLayout.PREFERRED_SIZE, 91, javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addContainerGap()

    );

    jPanelRespostaConsignLayout.setVerticalGroup(

jPanelRespostaConsignLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanelRespostaConsignLayout.createSequentialGroup()
        .addGap(6, 6, 6)
        .addComponent(jLabelTitleResposta)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanelRespostaConsignLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)

        .addComponent(jLabelRefId)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanelRespostaConsignLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jLabelCodigo)
        .addComponent(jLabelDescricao)))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanelRespostaConsignLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.
BASELINE)

        .addComponent(jTextFieldDescricao,
jPanelRespostaConsignLayout.createParallelGroup(javax.swing.GroupLayout.PREFERRED_SIZE,
jPanelRespostaConsignLayout.DEFAULT_SIZE,
jPanelRespostaConsignLayout.PREFERRED_SIZE)
        .addComponent(jTextFieldCodigo, javax.swing.GroupLayout.PREFERRED_SIZE,
jPanelRespostaConsignLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jTextFieldRefId, javax.swing.GroupLayout.PREFERRED_SIZE,
jPanelRespostaConsignLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

    );

    javax.swing.GroupLayout jPanelRespostaConsignLayout = new
    javax.swing.GroupLayout(jPanelRespostaConsignLayout.getContentPane());
    jPanelRespostaConsignLayout.getContentPane().setLayout(jPanelRespostaConsignLayout);
    jPanelRespostaConsignLayout.setLayout(jPanelRespostaConsignLayout);

    jPanelRespostaConsignLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanelRespostaConsignLayout.createSequentialGroup()
            .addContainerGap()

```

```

.addGroup(jInternalFrameWorkorderLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addComponent(jPanelConsign, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

    .addGroup(jInternalFrameWorkorderLayout.createSequentialGroup()

        .addComponent(jPanelRespostaConsign,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGap(0, 0, Short.MAX_VALUE)))

    .addContainerGap())

);

jInternalFrameWorkorderLayout.setVerticalGroup(

jInternalFrameWorkorderLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addGroup(jInternalFrameWorkorderLayout.createSequentialGroup()

        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

        .addComponent(jPanelConsign, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(jPanelRespostaConsign, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGap(206, 206, 206))

    );

jInternalFrameCheckStatus.setClosable(true);
jInternalFrameCheckStatus.setIconifiable(true);
jInternalFrameCheckStatus.setResizable(true);
jInternalFrameCheckStatus.setPreferredSize(new java.awt.Dimension(800, 40));
jInternalFrameCheckStatus.setVisible(true);
jInternalFrameCheckStatus.getContentPane().setLayout(null);

jLabelTitleCheckStatus.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
jLabelTitleCheckStatus.setText("Check Status");
jInternalFrameCheckStatus.getContentPane().add(jLabelTitleCheckStatus);
jLabelTitleCheckStatus.setBounds(10, 0, 130, 22);

jLabelWorkOrder1.setText("Workorder nº ");
jInternalFrameCheckStatus.getContentPane().add(jLabelWorkOrder1);
jLabelWorkOrder1.setBounds(20, 30, 110, 14);
jInternalFrameCheckStatus.getContentPane().add(jTextFieldWorkOrder1);
jTextFieldWorkOrder1.setBounds(20, 50, 90, 20);

jButtonCheckS.setText("Enviar");
jButtonCheckS.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButtonCheckSActionPerformed(evt);
    }
});

```

```
jInternalFrameCheckStatus.getContentPane().add(jButtonCheckS);
jButtonCheckS.setBounds(20, 80, 90, 23);

jTextArea1.setColumns(20);
jTextArea1.setRows(5);
jScrollPane3.setViewportViewView(jTextArea1);

jInternalFrameCheckStatus.getContentPane().add(jScrollPane3);
jScrollPane3.setBounds(150, 10, 170, 90);

jPanelRespostaConsign1.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.RAISED));

jLabelTitleRespostal.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
jLabelTitleRespostal.setText("Resposta");

jLabelRefIdCS.setText("Ref Id");

jLabelCodigoCS.setText("Código");

jLabelDescricaoCS.setText("Descrição");

javax.swing.GroupLayout jPanelRespostaConsign1Layout = new
javax.swing.GroupLayout(jPanelRespostaConsign1);
jPanelRespostaConsign1.setLayout(jPanelRespostaConsign1Layout);
jPanelRespostaConsign1Layout.setHorizontalGroup(
    jPanelRespostaConsign1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanelRespostaConsign1Layout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addGroup(jPanelRespostaConsign1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(jPanelRespostaConsign1Layout.createSequentialGroup()
                    .addComponent(jLabelRefIdCS, javax.swing.GroupLayout.PREFERRED_SIZE, 60, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(jTextFieldCodigoCS, javax.swing.GroupLayout.PREFERRED_SIZE, 131, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(20, 20, 20))
                .addGroup(jPanelRespostaConsign1Layout.createSequentialGroup()
                    .addComponent(jLabelDescricaoCS, javax.swing.GroupLayout.PREFERRED_SIZE, 60, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(jTextFieldDescricaoCS, javax.swing.GroupLayout.PREFERRED_SIZE, 131, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(20, 20, 20))
            )
        )
        .addGroup(jPanelRespostaConsign1Layout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addComponent(jLabelTituloResposta, javax.swing.GroupLayout.PREFERRED_SIZE, 100, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(10, true)
        )
    );
```

```
.addGroup(jPanelRespostaConsign1Layout.createSequentialGroup()  
    .addGap(109, 109, 109)  
    .addComponent(jLabelCodigoCS)  
  
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,  
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)  
    .addComponent(jLabelDescricaoCS))  
  
    .addGroup(jPanelRespostaConsign1Layout.createSequentialGroup()  
        .addComponent(jLabelRefIdCS)  
        .addGap(19, 19, 19)  
        .addComponent(jLabelTitleRespostal,  
javax.swing.GroupLayout.PREFERRED_SIZE, 180, javax.swing.GroupLayout.PREFERRED_SIZE)))  
        .addGap(0, 77, Short.MAX_VALUE)))  
        .addContainerGap()  
    );  
jPanelRespostaConsign1Layout.setVerticalGroup(  
  
jPanelRespostaConsign1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
    .addGroup(jPanelRespostaConsign1Layout.createSequentialGroup()  
        .addGap(6, 6, 6)  
        .addComponent(jLabelTitleRespostal)  
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  
  
    .addGroup(jPanelRespostaConsign1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
        .addComponent(jLabelRefIdCS)  
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,  
jPanelRespostaConsign1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)  
            .addComponent(jLabelCodigoCS)  
            .addComponent(jLabelDescricaoCS)))  
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  
  
    .addGroup(jPanelRespostaConsign1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
  
    .addGroup(jPanelRespostaConsign1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)  
        .addComponent(jTextFieldDescricaoCS,  
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE)  
        .addComponent(jTextFieldCodigoCS,  
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE))  
        .addComponent(jTextFieldRefIdCS, javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))  
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))  
    );
```

```
jInternalFrameCheckStatus.getContentPane().add(jPanelRespostaConsign1);
jPanelRespostaConsign1.setBounds(347, 10, 340, 89);

jScrollPane4.setAutoscrolls(true);

jTextAreaDebug.setColumns(20);
jTextAreaDebug.setRows(5);
jScrollPane4.setViewportViewView(jTextAreaDebug);

jScrollPane5.setAutoscrolls(true);

jTextAreaDebug1.setColumns(20);
jTextAreaDebug1.setRows(5);
jScrollPane5.setViewportViewView(jTextAreaDebug1);

jMenu1.setText("Menu");

jMItemConsign.setText("Consign Stock");
jMItemConsign.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMItemConsignActionPerformed(evt);
    }
});
jMItemConsign.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyPressed(java.awt.event.KeyEvent evt) {
        jMItemConsignKeyPressed(evt);
    }
});
jMenu1.add(jMItemConsign);

jMenuICheckStatus.setText("Check Status");
jMenuICheckStatus.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuICheckStatusActionPerformed(evt);
    }
});
jMenu1.add(jMenuICheckStatus);

jMenuBarNoBd.add(jMenu1);
jMenu1.setAccessibleContext().setAccessibleName("Gestão");

jMenu2.setText("Edit");
jMenuBarNoBd.add(jMenu2);

setJMenuBar(jMenuBarNoBd);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .add(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .add(jMenu1)
                .add(jMenu2)
            )
            .add(jMenuICheckStatus)
            .add(jMItemConsign)
        )
        .add(jMenu1)
        .add(jMenu2)
        .add(jMenuICheckStatus)
        .add(jMItemConsign)
    );
}
```

```

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jInternalFrameWorkorder,
javax.swing.GroupLayout.PREFERRED_SIZE, 974, javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGroup(layout.createSequentialGroup()
        .addGap(930, 930, 930)
        .addComponent(jScrollPane5, javax.swing.GroupLayout.PREFERRED_SIZE,
158, javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGroup(layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jInternalFrameCheckStatus,
javax.swing.GroupLayout.PREFERRED_SIZE, 720, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(18, 18, 18)
        .addComponent(jScrollPane4, javax.swing.GroupLayout.PREFERRED_SIZE,
158, javax.swing.GroupLayout.PREFERRED_SIZE)))
    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jInternalFrameWorkorder,
javax.swing.GroupLayout.PREFERRED_SIZE, 480, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jInternalFrameCheckStatus,
javax.swing.GroupLayout.PREFERRED_SIZE, 134, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jScrollPane4, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(34, 34, 34)
    .addComponent(jScrollPane5, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addContainerGap(282, Short.MAX_VALUE))
);

pack();
} // </editor-fold>

private void jMenuItemConsignActionPerformed(java.awt.event.ActionEvent evt) {

    jInternalFrameWorkorder.setVisible(true);
    jInternalFrameCheckStatus.setVisible(false);

    DocumentReferenceType checkStatus = DocReference.DocRef("FS000000099");//pedido
check status
    jTextFieldWorkOrder.setText("");

```



```

jTextFieldContaCliente.setText("");
jTextFieldAccessId.setText("");
jTextFieldNomeCliente.setText("");
jTextFieldContribuinteCliente.setText("");

jTextFieldCodigo.setText("");
jTextFieldRefId.setText("");
jTextFieldDescricao.setText("");

jTextAreaReceiptAdvice.setText("");

}

private void jMenuItemConsignKeyPressed(java.awt.event.KeyEvent evt) {

}

private void jButtonEnviarActionPerformed(java.awt.event.ActionEvent evt) {
    ReceiptAdviceType receiptAdvice = new ReceiptAdviceType();

    try {
        String itemid = "";
        for (int i = 0; i < jTableItens.getRowCount(); i++) {
            if ((jTableItens.getValueAt(i, 0)) != null)
            {
                itemid = jTableItens.getValueAt(i, 0).toString();
                itemID.add(i, itemid);
            }
            else{
                itemid = null;
                itemID.add(i, itemid);
            }
        }

        String receive = "";
        for (int i = 0; i < jTableItens.getRowCount(); i++) {
            if ((jTableItens.getValueAt(i, 1)) != null)
            {
                receive = jTableItens.getValueAt(i, 1).toString();
                Receivedqt.add(i, receive);
            }
            else {
                receive = null;
                Receivedqt.add(i, receive);
            }
        }

        String reject = "";
        for (int i = 0; i < jTableItens.getRowCount(); i++) {

```

```
        if ((jTableItens.getValueAt(i, 2)) != null)
        {
            reject = jTableItens.getValueAt(i, 2).toString();
            Rejectedqdt.add(i, reject);
        }
        else {
            reject = null;
            Rejectedqdt.add(i, reject);
        }
    }

    String material = "";
    for (int i = 0; i < jTableItens.getRowCount(); i++) {
        if ((jTableItens.getValueAt(i, 3)) != null)
        {
            material = jTableItens.getValueAt(i, 3).toString();
            Material.add(i, material);
        }

        else{
            material = null;
            Material.add(i, material);
        }
    }

    String serie = "";
    for (int i = 0; i < jTableItens.getRowCount(); i++) {
        if ((jTableItens.getValueAt(i, 4)) != null)
        {
            serie = jTableItens.getValueAt(i, 4).toString();
            Serie.add(i, serie);
        }
        else{
            serie = null;
            Serie.add(i, serie);
        }
    }

    copyIndicator.setValue(false);
    receiptAdvice.setCopyIndicator(copyIndicator);
    receiptAdvice.setUBLVersionID(UBL);

    UBL.setValue("2.0");

    String fsid = jTextFieldFsId.getText().toString();
    String workorder = jTextFieldWorkOrder.getText().toString();
    String uuID = jTextFieldUUID.getText().toString();
    idWorkorder.setValue(workorder);
```

```

        receiptAdvice.setID(idWorkorder);
        uuid.setValue(uuid);

        receiptAdvice.setUUID(uuid);
        receiptAdvice.getReceiptLine().addAll(RecLineList.ReceiptLineList(itemID,
Receivedqt, Rejectedqt, Material, Serie));
        CustomerPartyType DeliveryCustParty = DelCustParty.custDCP("337631456",
"Axentel", fsid, "504657038");
        SupplierPartyType DespatchSupParty = DespSupParty.custDSP("453476327",
"Vodafone", "Vod1", "502544580");
        String contaCliente = jTextFieldContaCliente.getText().toString();
        String accessId = jTextFieldAccessId.getText().toString();
        String nomeCliente = jTextFieldNomeCliente.getText().toString();
        String contribuinte = jTextFieldContribuinteCliente.getText().toString();
        CustomerPartyType BuyerCustParty = BuyCustParty.custBCP(contaCliente,
accessId, nomeCliente, contribuinte);
        receiptAdvice.setDeliveryCustomerParty(DeliveryCustParty);
        receiptAdvice.setDespatchSupplierParty(DespatchSupParty);
        receiptAdvice.setBuyerCustomerParty(BuyerCustParty);

        String mydatetime=jTextFieldData.getText().toString();
        XMLGregorianCalendar xgc = null;
    try {
        xgc = DatatypeFactory.newInstance().newXMLGregorianCalendar(mydatetime);
    } catch (DatatypeConfigurationException ex) {
        Logger.getLogger(JFrameNoBd.class.getName()).log(Level.SEVERE, null, ex);
    }

    issueDate.setValue(xgc);
    receiptAdvice.setIssueDate(issueDate);

    Authenticator.setDefault (new Authenticator() {
    protected PasswordAuthentication getPasswordAuthentication() {
        return new PasswordAuthentication ("xxxxxxxx", "xxxxxxxx".toCharArray());
    }
});

// Create a trust manager that does not validate certificate chains
TrustManager[] trustAllCerts = new TrustManager[]{
    new X509TrustManager() {
        public java.security.cert.X509Certificate[] getAcceptedIssuers() {
            return null;
        }

        public void checkClientTrusted(
            java.security.cert.X509Certificate[] certs, String authType) {
        }
    }
}

```

```

        public void checkServerTrusted(
            java.security.cert.X509Certificate[] certs, String authType) {
        }
    }
};

// Install the all-trusting trust manager
try {
    SSLContext sc = SSLContext.getInstance("SSL");

    sc.init(null, trustAllCerts, new java.security.SecureRandom());
    HttpsURLConnection.setDefaultSSLSocketFactory(sc.getSocketFactory());
} catch (Exception e) {
}

//      Conversão para XML

ObjectFactory Ofactory1 = new ObjectFactory();
JAXBContext context =
JAXBContext.newInstance(oasis.names.specification.ubl.schema.xsd.receiptadvice_2.ObjectFactor
y.class);

    JAXBElement<ReceiptAdviceType> element =
Ofactory1.createReceiptAdvice(receiptAdvice);

    Marshaller marshaller = context.createMarshaller();
    marshaller.setProperty("jaxb.formatted.output", Boolean.TRUE);
    StringWriter stringWriter = new StringWriter();
    marshaller.marshal(element, stringWriter);
    String xml = stringWriter.toString();
    jTextAreaReceiptAdvice.setText(xml);
    element = null;

    ResponseType RespConsignStock = new ResponseType();
    RespConsignStock = consignStock(receiptAdvice);
        jTextFieldRefId.setText(RespostaConsign.DocRef(RespConsignStock));
        jTextFieldCodigo.setText(RespostaConsign.CodRef(RespConsignStock));

jTextFieldDescricao.setText(RespostaConsign.DescricaoRef(RespConsignStock));

    } catch (JAXBException ex) {
        Logger.getLogger(JFrameNoBd.class.getName()).log(Level.SEVERE, null, ex);
        String excecao = ex.getMessage().toString();
        jTextAreaDebug.setText("JAB exceção"+excecao);
    }
}

private void jTextFieldAccessIdActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

```

```

    }

    private void jTextFieldContaClienteActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void jButtonCheckSActionPerformed(java.awt.event.ActionEvent evt) {

        ApplicationResponseType appResposta = new ApplicationResponseType();
        DocumentReferenceType pedidoCheckStatus = new DocumentReferenceType();

        String WoRef;
        WoRef = jTextFieldWorkOrder1.getText();

        IDType docreferenceId = new IDType();
        docreferenceId.setValue(WoRef);
        pedidoCheckStatus.setID(docreferenceId);

        Authenticator.setDefault (new Authenticator() {
        protected PasswordAuthentication getPasswordAuthentication() {
            return new PasswordAuthentication ("xxxxxxx", "xxxxxxx".toCharArray());
        }
    });

        // Create a trust manager that does not validate certificate chains
        TrustManager[] trustAllCerts = new TrustManager[]{
            new X509TrustManager() {
                public java.security.cert.X509Certificate[] getAcceptedIssuers() {
                    return null;
                }

                public void checkClientTrusted(
                    java.security.cert.X509Certificate[] certs, String authType) {
                }

                public void checkServerTrusted(
                    java.security.cert.X509Certificate[] certs, String authType) {
                }
            }
        };

        // Install the all-trusting trust manager
        try {
            SSLContext sc = SSLContext.getInstance("SSL");

            sc.init(null, trustAllCerts, new java.security.SecureRandom());
            HTTPSURLConnection.setDefaultSSLSocketFactory(sc.getSocketFactory());
        } catch (Exception e) {
        }

        appResposta = checkConsignStatus(pedidoCheckStatus);

        jTextFieldRefIdCS.setText(RespostaCheckStatus.DocRef(appResposta));
        jTextFieldRefIdCS.validate();
    }

```

```

        jTextFieldCodigoCS.setText(RespostaCheckStatus.CodRef(appResposta));
        jTextFieldDescricaoCS.setText(RespostaCheckStatus.DescricaoRef(appResposta));

    }

    private void
jInternalFrameWorkorderInternalFrameClosed(javax.swing.event.InternalFrameEvent evt) {

    }

    private void jMenuItemCheckStatusActionPerformed(java.awt.event.ActionEvent evt) {
        jInternalFrameCheckStatus.setVisible(true);
        jInternalFrameWorkorder.setVisible(false);
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code
(optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default
look and feel.
         * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

        }

        java.util.logging.Logger.getLogger(JFrameNoBd.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (InstantiationException ex) {

        }

        java.util.logging.Logger.getLogger(JFrameNoBd.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (IllegalAccessException ex) {

        }

        java.util.logging.Logger.getLogger(JFrameNoBd.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

        }

        java.util.logging.Logger.getLogger(JFrameNoBd.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    }
    //</editor-fold>

    /* Create and display the form */

```

```

        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new JFrameNoBd().setVisible(true);
            }
        });
    }
    // Variables declaration - do not modify
    private javax.swing.JButton jButtonCheckS;
    private javax.swing.JButton jButtonEnviar;
    private javax.swing.JInternalFrame jInternalFrameCheckStatus;
    private javax.swing.JInternalFrame jInternalFrameWorkorder;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabelAccessId;
    private javax.swing.JLabel jLabelCodigo;
    private javax.swing.JLabel jLabelCodigoCS;
    private javax.swing.JLabel jLabelContaCliente;
    private javax.swing.JLabel jLabelData;
    private javax.swing.JLabel jLabelDescricao;
    private javax.swing.JLabel jLabelDescricaoCS;
    private javax.swing.JLabel jLabelFsId;
    private javax.swing.JLabel jLabelNomeCliente;
    private javax.swing.JLabel jLabelRefId;
    private javax.swing.JLabel jLabelRefIdCS;
    private javax.swing.JLabel jLabelTitleCheckStatus;
    private javax.swing.JLabel jLabelTitleResposta;
    private javax.swing.JLabel jLabelTitleResposta1;
    private javax.swing.JLabel jLabelTitleWo;
    private javax.swing.JLabel jLabelUUID;
    private javax.swing.JLabel jLabelWorkOrder;
    private javax.swing.JLabel jLabelWorkOrder1;
    private javax.swing.JLabel jLabeldataexemplo;
    private javax.swing.JMenuItem jMenuItemConsign;
    private javax.swing.JMenu jMenu1;
    private javax.swing.JMenu jMenu2;
    private javax.swing.JMenuBar jMenuBarNoBd;
    private javax.swing.JMenuItem jMenuItemCheckStatus;
    private javax.swing.JPanel jPanelConsign;
    private javax.swing.JPanel jPanelRespostaConsign;
    private javax.swing.JPanel jPanelRespostaConsign1;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JScrollPane jScrollPane2;
    private javax.swing.JScrollPane jScrollPane3;
    private javax.swing.JScrollPane jScrollPane4;
    private javax.swing.JScrollPane jScrollPane5;
    private javax.swing.JTable jTableItens;
    private javax.swing.JTextArea jTextArea1;
    private javax.swing.JTextArea jTextAreaDebug;
    private javax.swing.JTextArea jTextAreaDebug1;

```

```

private javax.swing.JTextArea jTextAreaReceiptAdvice;
private javax.swing.JTextField jTextFieldAccessId;
private javax.swing.JTextField jTextFieldCodigo;
private javax.swing.JTextField jTextFieldCodigoCS;
private javax.swing.JTextField jTextFieldContaCliente;
private javax.swing.JTextField jTextFieldContribuinteCliente;
private javax.swing.JTextField jTextFieldData;
private javax.swing.JTextField jTextFieldDescricao;
private javax.swing.JTextField jTextFieldDescricaoCS;
private javax.swing.JTextField jTextFieldFsId;
private javax.swing.JTextField jTextFieldNomeCliente;
private javax.swing.JTextField jTextFieldRefId;
private javax.swing.JTextField jTextFieldRefIdCS;
private javax.swing.JTextField jTextFieldUUID;
private javax.swing.JTextField jTextFieldWorkOrder;
private javax.swing.JTextField jTextFieldWorkOrder1;
private java.awt.Menu menu1;
private java.awt.Menu menu2;
private java.awt.MenuBar menuBar1;
// End of variables declaration

private static ResponseType
consignStock(oasis.names.specification.ubl.schema.xsd.receiptadvice_2.ReceiptAdviceType
parameters) {
    pt.vodafone.consignstock.ConsignStock_Service service = new
pt.vodafone.consignstock.ConsignStock_Service();
    pt.vodafone.consignstock.ConsignStock port = service.getConsignStockSOAP();
    return port.consignStock(parameters);
}

private static ApplicationResponseType
checkConsignStatus(oasis.names.specification.ubl.schema.xsd.commonaggregatecomponents_2.Docu
mentReferenceType parameters) {
    pt.vodafone.consignstock.ConsignStock_Service service = new
pt.vodafone.consignstock.ConsignStock_Service();
    pt.vodafone.consignstock.ConsignStock port = service.getConsignStockSOAP();
    return port.checkConsignStatus(parameters);
}
}

```


Anexo B – Testes da Aplicação

Neste anexo é reproduzido o documento que serviu para testar a aplicação desenvolvida neste projeto.

VODAFONE PORTUGAL

Projecto	CPE Management				
Versão	1	Data	20-01-2012	Última Actualização por:	Rui Branco em 31-10-2013
Documento	Testes integrados Vodafone – CPE Management				
	Cenários de teste e resultados				

Testes a Efectuar

Teste 1 - Cenário de Consignação

- Teste 1.1 - só com números de série
- Teste 1.2 - só com quantidades
- Teste 1.3 - com números de série e quantidades

Teste 2 - Cenário de Desconsignação

- Teste 2.1 - total só com números de série
- Teste 2.2 - parcial só com números de série

- Teste 2.3 - total só com quantidades
- Teste 2.4 - parcial só com quantidades

- Teste 2.5 - total com números de série e quantidades
- Teste 2.6 - parcial com números de série e quantidades

Teste 3 - Cenário de troca de material

- Teste 3.1 - Registo de troca de material parcialmente
- Teste 3.2 - Registo de troca de material na totalidade

Teste 4 – Testes end to end

Teste 5 – Movimentos de material entre Empresas de Field Service

- Teste 5.1 – Desconsignação de material na totalidade com centro 8300
- Teste 5.2 – Desconsignação de material na totalidade com centro 8500
- Teste 5.3 – Desconsignação de material na totalidade com centro 8600
- Teste 5.4 – Desconsignação de material parcial com centro 8300
- Teste 5.5 – Desconsignação de material parcial com centro 8500
- Teste 5.6 – Desconsignação de material parcial com centro 8600
- Teste 5.7 – Troca de material parcial com centro 8300
- Teste 5.8 – Troca de material parcial com centro 8500
- Teste 5.9 – Troca de material parcial com centro 8600

TESTE 1.1	
TESTE / OBJECTIVO	Cenário de Consignação Consignação só com números de série
DADOS A UTILIZAR	<p>Cliente: Conta: 308368401/ ADSL ID: 108424053 / WO 4000001</p> <p>FS ID: WD01</p> <p>Material: 520466 / STB IPTV CISCO CIS5230 PVR Quantidade: 1 Número Série: CA255001631</p> <p>108003901/ TECHNICOLOR TG784n Quantidade: 2 Número Série: 0876FF8688E1 + 0876FF87457D</p>
RESULTADO FINAL	Código: 0000 Descrição: SUCCESS
Estado:	<input checked="" type="checkbox"/> OK <input type="checkbox"/> Teste não efectuado <input type="checkbox"/> Not OK Data do Teste: 06.11.2013
Observações	
Comentários:	

TESTE 1.2	
TESTE / OBJECTIVO	Cenário de Consignação Consignação só com quantidades
DADOS A UTILIZAR	<p>Cliente: Conta: 308956678 / ADSL ID: 158099694 / WO 4000002</p> <p>FS ID: WD03</p> <p>Material: 520490 / DEVOLO ADAPTER MODEL 200MBPS AVPLUS Quantidade: 2</p> <p>Material: 108003893/ RUWIDO COMANDO VERSAO 3 Quantidade: 2</p>
RESULTADO FINAL	<i>Código: 0000 Descrição: SUCCESS</i>
Estado:	<input checked="" type="checkbox"/> OK <input type="checkbox"/> Teste não efectuado <input type="checkbox"/> Not OK Data do Teste: 06.11.2013
Observações	
Comentários:	

TESTE 1.3	
TESTE / OBJECTIVO	Cenário de Consignação Consignação com números de série e quantidades
DADOS A UTILIZAR	<p>Cliente: Conta: 308460278 / ADSL ID: 117899091 / WO 4000003 FS ID: WD02</p> <p>Material: 108003893/ RUWIDO COMANDO VERSAO 3 Quantidade: 1</p> <p>Cliente: Conta: 308460278 / ADSL ID: 117899091 / WO 4000004 FS ID: WD03</p> <p>520466/ STB IPTV CISCO CIS5230 Quantidade: 1 Número Série: CA255001603 108003901/ TECHNICOLOR TG784n Quantidade: 1 Número Série: 0876FF882BCD 520490 / DEVOLO ADAPTER MODEL 200MBPS AVPLUS Quantidade: 2 108003893/ RUWIDO COMANDO VERSAO 3 Quantidade: 1</p>
RESULTADO FINAL	Código: 0000 Descrição: SUCCESS
Estado:	<input checked="" type="checkbox"/> OK <input type="checkbox"/> Teste não efectuado <input type="checkbox"/> Not OK Data do Teste: 06.11.2013
Observações	
Comentários:	Foi feito o teste com a WO 4000004, a WO 4000003 deu erro.

TESTE 2.1	
TESTE / OBJECTIVO	Cenário de Desconsignação Desconsignação TOTAL só com números de série
DADOS A UTILIZAR	Cliente: Conta: / ADSL ID: / WO FS ID: Material: / Quantidade: Número Série: / Quantidade: Número Série:
RESULTADO FINAL	
Estado:	<input type="checkbox"/> OK <input type="checkbox"/> Teste não efectuado <input type="checkbox"/> Not OK Data do Teste:
Observações	
Comentários:	

TESTE 2.2	
TESTE / OBJECTIVO	Cenário de Desconsignação Desconsignação PARCIAL só com números de série
DADOS A UTILIZAR	Cliente: Conta: / ADSL ID: / WO FS ID: Material: / Quantidade: Número de série:
RESULTADO FINAL	
Estado:	<input type="checkbox"/> OK <input type="checkbox"/> Teste não efectuado <input type="checkbox"/> Not OK Data do Teste:
Observações	
Comentários:	

TESTE 2.3	
TESTE / OBJECTIVO	Cenário de Desconsignação Desconsignação TOTAL só com quantidades
DADOS A UTILIZAR	<p>Cliente: Conta: / ADSL ID: / WO</p> <p>FS ID:</p> <p>Material: / Quantidade:</p>
RESULTADO FINAL	
Estado:	<input type="checkbox"/> OK <input type="checkbox"/> Teste não efectuado <input type="checkbox"/> Not OK Data do Teste:
Observações	
Comentários:	

TESTE 2.4	
TESTE / OBJECTIVO	Cenário de Desconsignação Desconsignação PARCIAL só com quantidades
DADOS A UTILIZAR	Cliente: Conta: / ADSL ID: / WO FS ID: Material: / Quantidade: Material: / Quantidade:
RESULTADO FINAL	
Estado:	<input type="checkbox"/> OK <input type="checkbox"/> Teste não efectuado <input type="checkbox"/> Not OK Data do Teste:
Observações	
Comentários:	

TESTE 2.5	
TESTE / OBJECTIVO	Cenário de Desconsignação Desconsignação TOTAL com números de série e quantidades
DADOS A UTILIZAR	Cliente: Conta: / ADSL ID: / WO FS ID: Material: / Quantidade: Número de série:
RESULTADO FINAL	
Estado:	<input type="checkbox"/> OK <input type="checkbox"/> Teste não efectuado <input type="checkbox"/> Not OK Data do Teste:
Observações	
Comentários:	

TESTE 2.6	
TESTE / OBJECTIVO	Cenário de Desconsignação Desconsignação PARCIAL com números de série e quantidades
DADOS A UTILIZAR	Cliente: Conta: / ADSL ID: / WO FS ID: Material: / Quantidade: Número Série: Material: / Quantidade:
RESULTADO FINAL	
Estado:	<input type="checkbox"/> OK <input type="checkbox"/> Teste não efectuado <input type="checkbox"/> Not OK Data do Teste:
Observações	
Comentários:	

TESTE 3.1	
TESTE / OBJECTIVO	Cenário de troca de material Registo de troca a Cliente – Troca de material PARCIAL
DADOS A UTILIZAR	<p>Cliente: Conta: / ADSL ID: / WO</p> <p>FS ID:</p> <p>Entrega: Material: / Quantidade: Número Série: Material: / Quantidade: Número Série: Material: / Quantidade:</p> <p>Recolha: Material: / Quantidade: Número Série: Material: / Quantidade:</p>
RESULTADO FINAL	
Estado:	<input type="checkbox"/> OK <input type="checkbox"/> Teste não efectuado <input type="checkbox"/> Not OK Data do Teste:
Observações	
Comentários:	

TESTE 3.2	
TESTE / OBJECTIVO	Cenário de troca de material Registo de troca a Cliente – Troca de material TOTAL
DADOS A UTILIZAR	<p>Cliente: Conta: / ADSL ID: / WO</p> <p>FS ID:</p> <p>Entrega: Material: / Quantidade: Número Série: Material: / Quantidade: Número Série: Material: / Quantidade:</p> <p>Recolha: Material: / Quantidade: Número Série: Material: / Quantidade: Número Série: Material: / Quantidade:</p>
RESULTADO FINAL	
Estado:	<input type="checkbox"/> OK <input type="checkbox"/> Teste não efectuado <input type="checkbox"/> Not OK Data do Teste:
Observações	
Comentários:	

TESTE 4	
TESTE / OBJECTIVO	Criação de processo <i>end-to-end</i> com o mesmo número de série.
DADOS A UTILIZAR	<p>Passo 1 – Consignar Cliente: Conta: / ADSL ID: / WO FS ID: WD03 Material: 108003656 / HUAWEI ONT HG8247a Quantidade: 1 Número Série: 0019C7A4E518 Material: 520490 / DEVOLO ADAPTER MODEL 200MBPS AVPLUS Quantidade: 1 Material: 108007997/ PHILIPS TELFIX DECT D1501B Quantidade: 1 Número Série: GD011310108253</p> <p>Passo 2 – Desconsignar Cliente: Conta: / ADSL ID: / WO FS ID: WD02 Material: 108003656 / HUAWEI ONT HG8247a Quantidade: 1 Número Série: 0019C7A4E518 Material: 108007997/ PHILIPS TELFIX DECT D1501B Quantidade: 1 Número Série: GD011310108253</p> <p>Passo 3 – Consignar a outro Cliente Cliente: Conta: / ADSL ID: / WO FS ID: WD02 Material: 108003656 / HUAWEI ONT HG8247a Quantidade: 1 Número Série: 0019C7A4E518 Material: 108007997/ PHILIPS TELFIX DECT D1501B Quantidade: 1 Número Série: GD011310108253 108003893/ RUWIDO COMANDO VERSAO 3 Quantidade: 1</p> <p>Passo 4 – Trocar ao Cliente Cliente: Conta: / ADSL ID: / WO FS ID: WD03</p> <p>Entregue Material: 108003656 / HUAWEI ONT HG8247a Quantidade: 1 Número Série: 0019C7A51BB4 Material: 108007997/ PHILIPS TELFIX DECT D1501B Quantidade: 1 Número Série: GD011310108254</p> <p>Recolhido Material: 108003656 / HUAWEI ONT HG8247a Quantidade: 1 Número Série: 0019C7A4E518</p>

	Material: 108007997/ PHILIPS TELFIX DECT D1501B Quantidade: 1 Número Série: GD011310108253
RESULTADO FINAL	
Estado:	<input type="checkbox"/> OK <input type="checkbox"/> Teste não efectuado <input type="checkbox"/> Not OK Data do Teste:
Observações	
Comentários:	

TESTE 5.1	
TESTE / OBJECTIVO	Movimentos de material entre Empresas de Field Service Desconsignação de material na totalidade com centro 8300
DADOS A UTILIZAR	Cliente: Conta: / ADSL ID: / WO Material: / Quantidade: Número Série: Material: / Quantidade:
RESULTADO	
Estado:	<input type="checkbox"/> OK <input type="checkbox"/> Teste não efectuado <input type="checkbox"/> Not OK Data do Teste:
Not OK: Observações	
Comentários:	

TESTE 5.2	
TESTE / OBJECTIVO	Movimentos de material entre Empresas de Field Service Desconsignação de material na totalidade com centro 8500
DADOS A UTILIZAR	Cliente: Conta: / ADSL ID: / WO Material: / Quantidade: Número Série: Material: / Quantidade:
RESULTADO	
Estado:	<input type="checkbox"/> OK <input type="checkbox"/> Teste não efectuado <input type="checkbox"/> Not OK Data do Teste:
Not OK: Observações	
Comentários:	

TESTE 5.3	
TESTE / OBJECTIVO	Movimentos de material entre Empresas de Field Service Desconsignação de material na totalidade com centro 8600
DADOS A UTILIZAR	Cliente: Conta: / ADSL ID: / WO Material: / Quantidade: Número Série: Material: / Quantidade:
RESULTADO	
Estado:	<input type="checkbox"/> OK <input type="checkbox"/> Teste não efetuado <input type="checkbox"/> Not OK Data do Teste:
Not OK: Observações	
Comentários:	

TESTE 5.4	
TESTE / OBJECTIVO	Movimentos de material entre Empresas de Field Service Desconsignação de material parcial com centro 8300
DADOS A UTILIZAR	Cliente: Conta: / ADSL ID: / WO Material: / Quantidade: Número Série: Material: / Quantidade:
RESULTADO	
Estado:	<input type="checkbox"/> OK <input type="checkbox"/> Teste não efectuado <input type="checkbox"/> Not OK Data do Teste:
Not OK: Observações	
Comentários:	

TESTE 5.5	
TESTE / OBJECTIVO	Movimentos de material entre Empresas de Field Service Desconsignação de material parcial com centro 8500
DADOS A UTILIZAR	Cliente: Conta: / ADSL ID: / WO Material: / Quantidade: Número Série: Material: / Quantidade:
RESULTADO	
Estado:	<input type="checkbox"/> OK <input type="checkbox"/> Teste não efectuado <input type="checkbox"/> Not OK Data do Teste:
Not OK: Observações	
Comentários:	

TESTE 5.6	
TESTE / OBJECTIVO	Movimentos de material entre Empresas de Field Service Desconsignação de material parcial com centro 8600
DADOS A UTILIZAR	Cliente: Conta: / ADSL ID: / WO Material: / Quantidade: Número Série: Material: / Quantidade:
RESULTADO	
Estado:	<input type="checkbox"/> OK <input type="checkbox"/> Teste não efectuado <input type="checkbox"/> Not OK Data do Teste:
Not OK: Observações	
Comentários:	

TESTE 5.7	
TESTE / OBJECTIVO	Movimentos de material entre Empresas de Field Service Troca de material parcial com centro 8300
DADOS A UTILIZAR	Cliente: Conta: / ADSL ID: / WO Material: Quantidade: Número Série: Material: / Quantidade:
RESULTADO	
Estado:	<input type="checkbox"/> OK <input type="checkbox"/> Teste não efectuado <input type="checkbox"/> Not OK Data do Teste:
Not OK: Observações	
Comentários:	

TESTE 5.8	
TESTE / OBJECTIVO	Movimentos de material entre Empresas de Field Service Troca de material parcial com centro 8500
DADOS A UTILIZAR	Cliente: Conta: 308712577/ ADSL ID: 155627395/ WO 2301127 Material: / Quantidade: Número Série: Material: / Quantidade:
RESULTADO	
Estado:	<input type="checkbox"/> OK <input type="checkbox"/> Teste não efectuado <input type="checkbox"/> Not OK Data do Teste:
Not OK: Observações	
Comentários:	

TESTE 5.9	
TESTE / OBJECTIVO	Movimentos de material entre Empresas de Field Service Troca de material parcial com centro 8600
DADOS A UTILIZAR	Cliente: Conta: / ADSL ID: / WO Material: / Quantidade: Número Série: Material: / Quantidade:
RESULTADO	
Estado:	<input type="checkbox"/> OK <input type="checkbox"/> Teste não efectuado <input type="checkbox"/> Not OK Data do Teste:
Not OK: Observações	
Comentários:	